Todospectrum

Diciembre 1984 250 Ptas.

AÑO 1 - NUMERO 4.

REVISTA EXCLUSIVA PARA USUARIOS



Consola para el Spectrum

Primeras instrucciones en Código Máquina

Contabilidad profesional

Programas: Cubo de Rubik, Archivos en microdrive, etc.

EXTRA: Programa- calendario 1985

Llega la divers

¡Participa y gana! ¡T

J. M. PUBLICIDAD

UN JUEGO PARA DISFRUTAR CON TUS AMIGOS

Con tu Spectrum (no importa el modelo, 16 ó 48 K.) y la cinta del "Un, Dos, Tres..." podrás conseguir diversión sin límites junto con los tuyos.

Por primera vez se te ofrece un concurso participativo para microordenador, donde pueden competir hasta tres parejas.

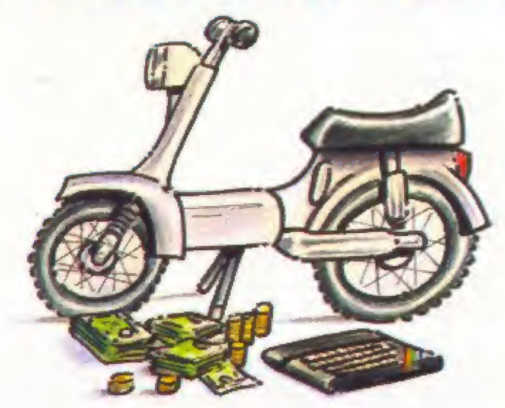
Y con la novedad de que el juego será cada vez distinto, pues siempre que conectes la cinta al Spectrum, te ofrecerá diferentes alternativas.



Concursa en casa con los protagonistas de la "tele".

Bigote Arrocet te invita en persona a un concurso exactamente igual que el de la "tele".
Basta con que pongas la cinta en tu cassette y él te irá dando las instrucciones de este nuevo y diferente juego.
Puedes conseguir premios tan sorprendentes como los del programa de Chicho y, por supuesto: también tiene Chollo.

1 de cada 3 cassetes contiene ¡PREMIO DIRECTO!:



Viajes, moto-vespas, sintetizadores, microordenadores Spectrum, QL, un montón de premios más, **y el gran Chollo:**



Para este número habíamos reservado amplio espacio a las novedades que se suponía íbamos a ver en la feria del SIMO. Dispersados por cuatro pabellones, poco nuevo ofrecieron las casas comerciales para quienes su público principal quedaba fuera del recinto ante la prohibición de la entrada a menores de 17 años. Una vez más se evidenciaba la necesidad de contar con ferias específicas para los ordenadores domésticos.

Como anunciábamos en el número anterior, comenzamos con la serie sobre el lenguaje PASCAL, ante las cartas recibidas en este sentido. Esperamos que siga incrementándose la comunicación para hacer de TODOSPECTRUM una revista participativa.

Y como las fechas mandan, además de Santaclair, en las páginas centrales encontrará un «montaje» muy particular de nuestra mascota. Un programa bien estructurado y explicado para que anote sus compromisos y obtenga el calendario del próximo año o del que desee.

Finalmente, como podrá ver, hemos incorporado a «nuestra plantilla» una nueva impresora para facilitar la introducción de los programas, que utilizaremos íntegramente a partir del próximo número.

Hasta el próximo año.



Cargado de Spectrums, Santaclair no faltó a la cita de estas navidades. Santiago Morga y Stnacho, del equipo de diseño de Dinamic Soft, lo hicieron posible trabajando con el programa Artist.

DIRECTOR:
Simeón Cruz
COORDINADOR EDITORIAL:
J. A. Sanz
REDACCION:
Juan Arencibia, Fernando García, José C. Tomás, Gumersindo García, Luis M. Brugarolas, Ricardo García.
DISEÑO:
Ricardo Segura
FOTOGRAFIA:
Ernesto Walfisch
Editado por:
Publinformática, S. A.
PRESIDENTE:
Fernando Bolín
DIRECTOR EDITORIAL:
Norberto Gallego
ADMINISTRACION:
Infodis, S. A.
CONSEJERO DELEGADO:
Fernando Bolín
GERENTE CIRCULACION Y VENTAS:
Luis Carrero
PRODUCCION:
Miguel Onieva
SERVICIO CLIENTES:
Antonio Zurdo
JEFE DE PUBLICIDAD:
María José Martín

Dirección, Redacción y Administración:
C/ Bravo Murillo, 377
Tel. 733 74 13
28020 Madrid
Publicidad Madrid: Nieves Clemente
C/ Bravo Murillo, 377
Tel. 733 96 62
28020 Madrid
Publicidad Barcelona: Mari Carmen Ríos
C/ Pelayo, 12
Tel. (93) 301 47 00 ext. 27
08001 Barcelona
Depósito Legal: M-29041-1984
Distribuye S.G.E.L.
Avda. Valdelaparra, s/n
Alcobendas (Madrid)
Fotomecánica: Karmat
C/ Pantoja, 10
Madrid
Imprime: Héroes
C/ Torrelara, 8
28036 Madrid

SUSCRIPCIONES:
Rogamos dirijan toda la correspondencia relacionada con suscripciones a:
TODOSPECTRUM
EDISA: Tel. 415 97 12
C/ López de Hoyos, 141-5.º
28002 MADRID
(Para todos los pagos reseñar solamente TODOSPECTRUM)

Para la compra de ejemplares atrasados diríjanse a la propia editorial
TODOSPECTRUM
C/ Bravo Murillo, 377 - 5.º A
Tel. 733 74 13
28020 MADRID



De profesión: PROGRAMADOR

Aunque nunca se haya parado a pensarlo, detrás de cada videojuego hay muchas horas de trabajo, y en muchos casos detrás de ellos encontrará los nombres de Joan Sales y Oscar Domingo, dos programadores con «solera» para los que el lenguaje máquina no representa ninguna dificultad.

«Nos hemos dado cuenta de que las máquinas comerciales grandes tienden a lo deportivo y el videojuego siempre es una competición contra algo», nos decía Joan. «Entonces o vamos a los juegos inteligentes, de lucha contra la máquina, o simular con la máquina unas condiciones lo más reales posibles. Una vez que se tiene clara la idea, hay que informarse sobre el tema, prescindiendo del aspecto informático. Así en el juego de esquí, estudiamos un manual y libros de física que nos ayudan a compensar las fuerzas que inciden sobre la persona que está esquiando. El paso inmediato son los algoritmos de comportamiento del esquiador».

Una vez estructurado el programa y resueltos los algoritmos de movimiento es cuando se empieza a pensar en el Spectrum, como nos indicaba Joan. «Hay que estudiar las posibilidades del Spectrum para llevar a cabo la idea: el color, la pantalla... El problema de codificar el programa es lo de menos. Puede contarse con el algoritmo necesario, pero puede resultar muy lento, incluso en código máquina. El movimiento del esquí en tres dimensiones, por ejemplo, es un cálculo trigonométrico constante y el código máquina del Spectrum no puede hacer estos cálculos a una velocidad aceptable».

Se dice que cuando se han realizado muchos programas la programación se convierte en una mezcolanza de rutinas. «En absoluto, los programas son siempre diferentes», replicaba Joan, «hay mucha competitividad y las técnicas empleadas en un juego, generalmente son obsoletas para el siguiente. Por ejemplo, ahora estoy con un algoritmo para trabajar en tres dimensiones con posibilidades de giro. Esto se consigue tratando los puntos en coordenadas polares en lugar de cartesianas, es decir, por ángulo y distancia a un origen. Como el Spectrum trabaja con coordenadas X, Y, hay que partir de cero y empezar a buscar rutinas sin trigonométrica, sin raíces cuadradas, sin decimales, debido al código máquina».

«Una limitación a tener en cuenta antes de meterte con el juego es la pantalla», proseguía Joan. «Es muy grande y está organizada para ser tratada fácilmente por la ULA. Manejarla desde código máquina no es un problema desde el punto de vista teórico, pero son 7K que hay que tratar de forma inteligente si no se quiere perder mucha velocidad, es decir, hay que hacer trucos para mover la pantalla mucho menos de lo que parece. El algoritmo no es problema, pero cuando el ordenador sabe lo que tiene que pintar, el acto de pintarlo se lleva hasta el 75 por ciento del tiempo. En definitiva es una técnica cinematográfica, para que la imagen no vaya a saltos, tiene que cambiar la pantalla ocho veces por segundo y según qué programa se quiera hacer, no siempre se puede conseguir.

Si las limitaciones del Spectrum lo permiten, el siguiente paso es el reparto del trabajo en bloques: impresión en pantallas, movimiento, sonidos, etc. Es aquí cuando se empieza a codificar y cuando otro ordenador entra en escena: El IBM-PC. «Pretendemos llenar el Spectrum totalmente de código máquina. Si metiésemos el objeto no habría sitio para el fuente que acostumbra a ser diez veces más largo. Para ahorrar memoria, los gráficos

están codificados y estos códigos vuelven a estar codificados en otros códigos más reducidos. Para esto tenemos programas estándar que nos facilitan mucho el trabajo, pero no podríamos llenar el Spectrum del código objeto sin la ayuda del PC. Otra ventaja adicional es la posibilidad de editar de forma fácil y rápida cualquier error, independientemente del teclado. Incluso en las teclas programables hemos colocado las instrucciones assembler más usuales para ganar tiempo en el teclado».

Una vez que el programa está completo, pasa al Spectrum a una velocidad de 16.000 baudios. «Compramos el PC por la amplia documentación técnica que traía, gracias a lo cual hicimos el circuito, el programa de recepción al Spectrum, otro de transmisión al PC y un protocolo para que los dos se entendiesen». El problema de las comunicaciones estaba resuelto para Joan. Ahora quedaba lo más tedioso. «Una vez cargado en el Spectrum, hay que probar el juego en todas sus posibilidades. El programador no puede hacer nunca este test, porque inconscientemente no hace las barbaridades que puede hacer quien no conoce el juego. A nosotros los errores no nos salen». Aunque reconocía que los períodos de pruebas no eran suficientemente exhaustivos. «La parte comercial mete prisa. Hay independencia, pero somos conscientes de que hay que colocarlo en el mercado. Además, el número de combinaciones posibles es prácticamente ilimitado y no se puede probar todo. Siempre hay algún error. Ahora estamos desarrollando una nueva técnica, consistente en realizar código máquina que se programase en función de situaciones. Normalmente las rutinas cuya función es programar a otras rutinas en función de determinadas condiciones. Esto complica mucho las cosas, aparte del tiempo que se lleva hacer todo esto».


«Oscar Domingo concentrado en el código máquina.»

«No nos enteramos de la comercialización, nosotros sólo programamos», responden ante la pregunta de compensación por el esfuerzo realizado. «Hay un buen apoyo en material de todo tipo: libros, videos, ordenadores, etc., pero se podía hacer más».



Con este artículo presentamos la descripción de una consola para tu Spectrum que podrá albergar, de forma cómoda, la impresora ZX PRINTER y un amplificador de sonido, haciendo mucho más cómoda la tarea de programación, y de utilización del ordenador en general.

CONSOLA SPECTRUM

LA consola permite que el Spectrum quede con el teclado ligeramente inclinado,lo cual es uno de los puntos esenciales para la comodidad del programador a la hora de utilizar el teclado.

El interruptor de encendido se encuentra situado en un lateral de la consola, de forma que queda empotrado en el lateral derecho, para que no sea demasiado accesible, y así evitar, en lo posible, que una mano demasiado curiosa produzca un desafortunado *reset* en el ordenador.

Otra de las grandes ventajas de la utilización de una consola, es el hecho de poder tener todo integrado, obteniendo un equipo más compacto y seguro, evitando así el típico lío de cables detrás del ordenador.

La impresora se ha dispuesto de forma que quede al nivel del teclado, y ligeramente desplazada a la derecha del panel de mandos, de forma que no entorpezca la mano derecha del programador al teclear.

Al quedar los conectores del Spectrum perpetuamente conectados, ya que sólo se utilizan los conmutadores e interruptores de la consola, los conectores del Spectrum no sufren, pudiéndose éstos desconectar cuando se quiera liberar el Spectrum de la consola, lo cual es inmediato.

Tras el encendido del ordenador con el interruptor empotrado en el lateral derecho se enciende un testigo de alimentación. Este testigo se encuentra situado en la esquina superior izquierda del ordenador. Como piloto se ha utilizado un diodo LED, como suele ser ya costumbre en estos casos.

En el panel de la impresora tenemos una serie de mandos que son, de izquierda a derecha:

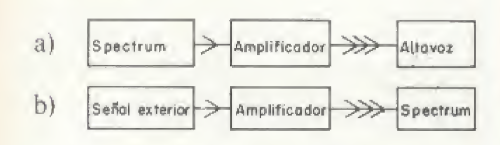
1) El indicador del conmutador de grabación SAVE-LOAD.

2) El conmutador de grabación SAVE-LOAD. Este conmutador facilita la tarea de salvar programas, ya que en la mayoría de los *cassettes* se produce un acople o realimentación si están los dos cables de EAR y de MIC conectados simultáneamente y con el *cassette* en posición de grabación, lo cual obliga a desconectar el cable de EAR cada vez que se va a salvar un programa. Bastará pues a partir de ahora con poner el conmutador en posición de SAVE antes de salvar un programa, sin necesidad de empezar a conectar y desconectar cables. Automáticamente, al poner el conmutador en posición de SAVE, se encenderá el LED testigo de SAVE, situado a la izquierda del conmutador.

3) Este conmutador invierte el sentido de amplificación del amplificador interno de la consola. Es decir, en posición normal, el amplificador funciona amplificando los sonidos producidos por el Spectrum y haciéndolos audibles en el altavoz integrado en la consola. Ahora bien, en la otra posición el amplificador recibe como señal de entrada a amplificar cualquier señal introducida en el conector de señal de entrada situado en la parte trasera de la consola, para entregarla una vez amplificada al Spectrum por la entrada de EAR. Esto

es muy útil cuando se quiere hacer análisis de sonidos con el ordenador. Podremos, por ejemplo, introducir la señal de un micrófono a la consola, y una vez amplificada, el Spectrum, mediante un programa adecuado, visualizará la frecuencia de la señal introducida.

Tenemos, en conclusión, estos dos casos:



4) El siguiente conmutador conecta o desconecta el altavoz, ya que es posible que nos interese tener el amplificador funcionando, pero sin que suene el altavoz.

5) Este botón es el control de volumen del amplificador, que se efectúa mediante un potenciómetro de 10 kilohomios logarítmico y con interruptor.

En la parte posterior de la consola se encuentra situado el interruptor del ventilador. No es demasiado importante incluir el ventilador en la consola ya que al quedar el Spectrum completamente levantado, el ordenador no se calienta demasiado, y sólo llega a templarse un poco la carcasa inferior.

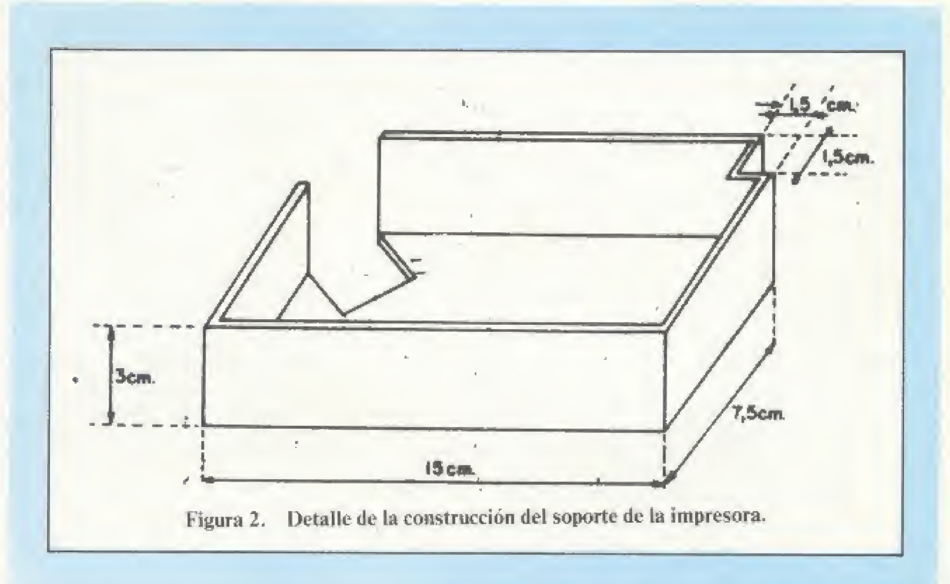


Figura 2. Detalle de la construcción del soporte de la impresora.

Construcción de la consola

Empezaremos por cortar enmadera de contrachapado de 3 milímetros, las tablas que constituyen la base y los costados laterales y frontales. Los ensamblamos con cola blanca y pegamos listones de refuerzo en las esquinas. Para esto nos fijaremos en la figura 1, en donde se encuentra desglosada la construcción mecánica de la consola.

Cortamos las piezas que constituyen el soporte para la impresora (ver fig. 2), y una vez construido lo pegamos al panel de mando, para después pegar éste a la consola. Una vez acabado esto, pegaremos los listones que servirán de soportes al ordenador en la consola y las últimas piezas de madera, para finalizar la construcción mecánica de la consola. Una vez que esté todo perfectamente seco, se lijará

* Cotas en centímetros

Figura 1. Dimensiones de la consola.

para quitar las imperfecciones y asperezas de la madera. Por último, se pintará con dos o tres capas de pintura para lacar. Por ejemplo, Titanlak. Una vez ya acabado todo el montaje mecánico de la consola y efectuados los taladros para la posterior fijación de los conmutadores, *leds* y el potenciómetro, construiremos el amplificador (ver montaje del amplificador).

Pegaremos las bandas de goma sobre las cuales se apoya el Spectrum, que van sobre los listones de sujeción.

Ahora comenzaremos a montar todos los componentes de la consola. Alojaremos el amplificador bajo la caja que soporta a la impresora. Montamos el potenciómetro, los tres conmutadores, y los dos diodos *leds*, con los cables que van a éstos ya soldados, pues una vez fijados a la consola resultaría muy difícil cablearlos. Fijamos el altavoz, y atornillamos la rejilla que tapará la membrana. Por fin empezaremos a cablear los distintos componentes de la consola, según el esquema del cableado de la figura 3.

Las resistencias que van conectadas en serie con los diodos LED son de 680 ohmios y de una potencia de 1/4 de watio. Si se quiere que luzcan más, se puede bajar el valor de la resistencia, por ejemplo a 470 ohmios. La resistencia que va conectada al interruptor del altavoz es de 8 ohmios y de 1 watio, y su función es de no dejar sin carga el amplificador en caso de desconectar el altavoz, que puede ser cualquiera de 4 u 8 ohmios de impedancia y de 1 watio como mínimo.

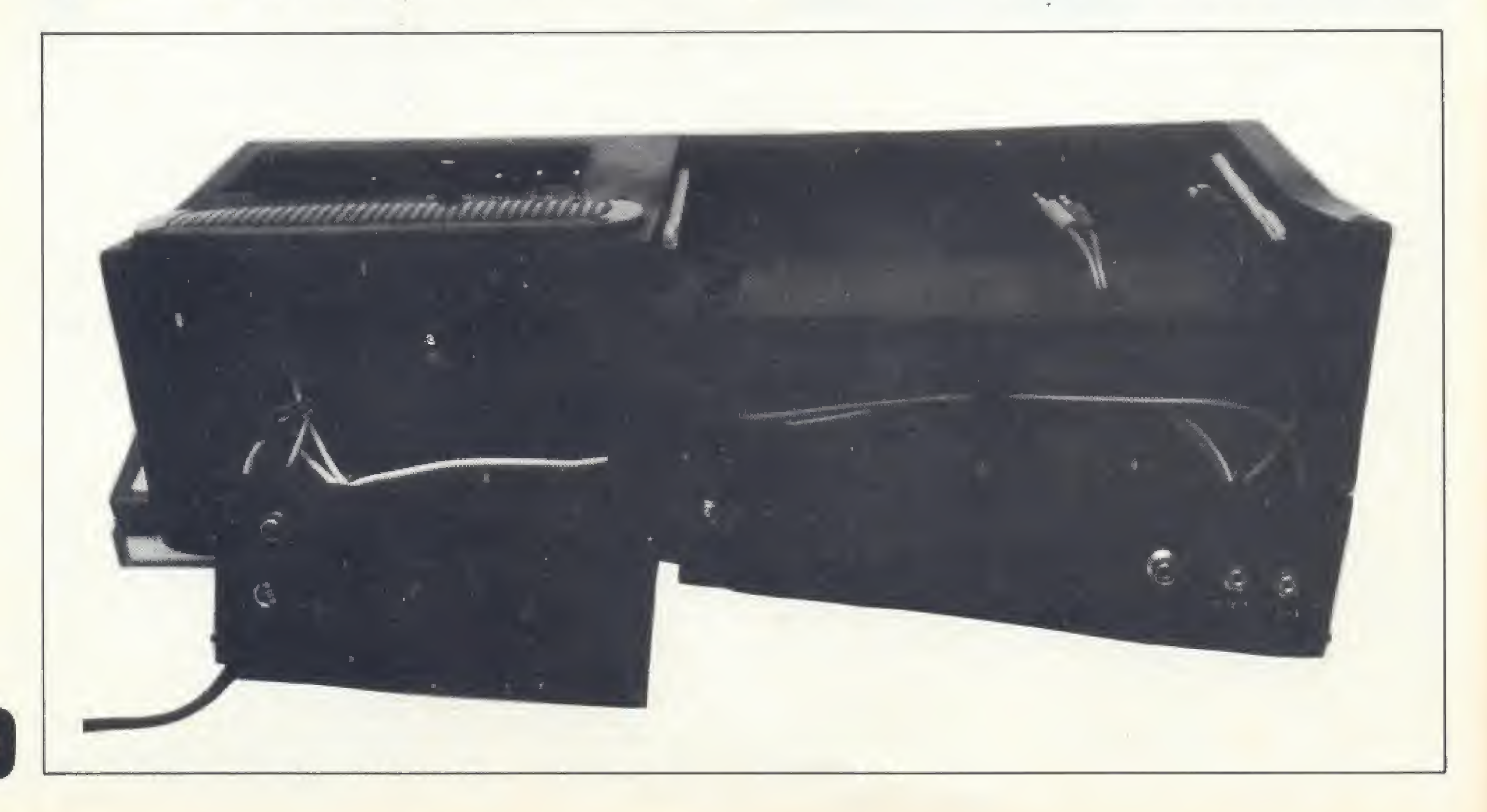
Prestar especial atención en soldar todo correctamente, aislando perfectamente todos los cables que estén a 220 voltios. Soldar los diodos LED, teniendo en cuenta su polaridad, ya que una inversión de ésta lo destruiría. Para ello fijarse en la parte plana que tiene en su base. El terminal más cercano a ésta corresponde al cátodo del LED, que es el que va conectado en serie con la resistencia limitadora al negativo de la alimentación. Los LEDs utilizados en la consola son de color rojo, aunque naturalmente se pueden utilizar verdes o naranjas. La conexión entre los conectores macho y hembra de antena se efectuará con cable coaxial del tipo empleado para las conexiones de antena de los televisores. Igualmente, las conexiones de los conectores EAR y MIC se efectuarán con cable coaxial del tipo empleado para baja frecuencia.

Para cualquier detalle del montaje, fijarse bien en las fotografías de la consola, lo cual nos puede ayudar a salir de alguna duda que surja durante el montaje. Para finalizar, indicamos que todo el material es perfectamente asequible en los comercio del ramo, y que el coste de la consola es de alrededor de las 2.000 ptas. en componentes electrónicos y material eléctrico. A esto le habremos de sumar el precio de las maderas de contrachapado, pegamento y pintura.

Amplificador de sonido para la consola

El amplificador utilizado en la consola para el Spectrum está basado en el circuito integrado TDA 2002 que proporciona una potencia de salida de cerca de 2 watios con un altavoz de 8 ohmios. Se pueden obtener potencias mayores utilizando un altavoz de menor impedancia, por ejemplo de 4 ohmios, aunque con 8 ohmios resulta satisfactorio el volumen obtenido.

El circuito integrado viene interiormente protegido contra cortocircuitos en su salida, sobretemperaturas, apertura del circuito respecto a masa e inversión de polaridad, lo cual le hace muy apto para



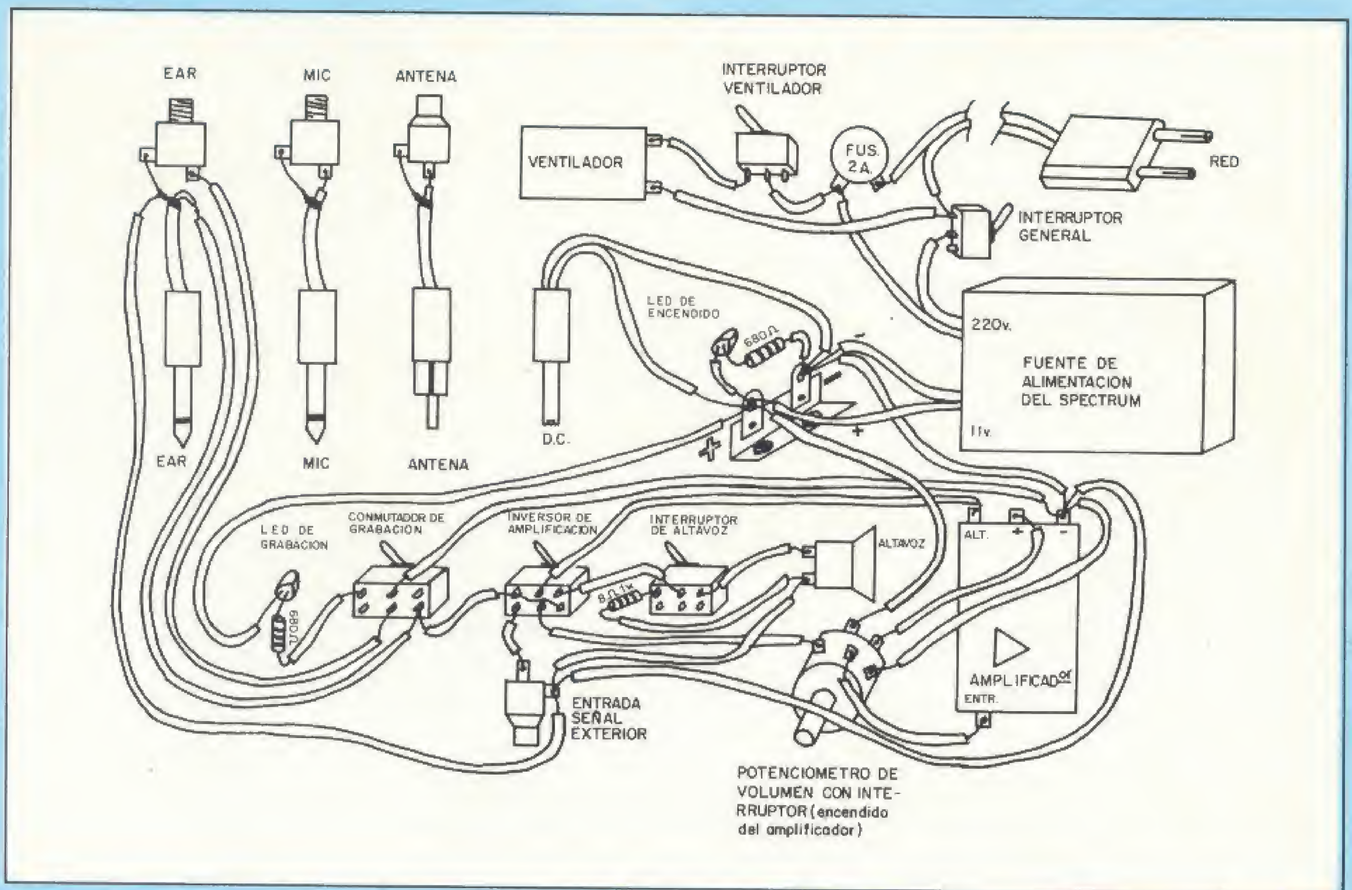


Figura 3. Cableado de la consola.

Figura 5. Detalle del montaje del radiador y patillaje del circuito integrado TDA 2002.

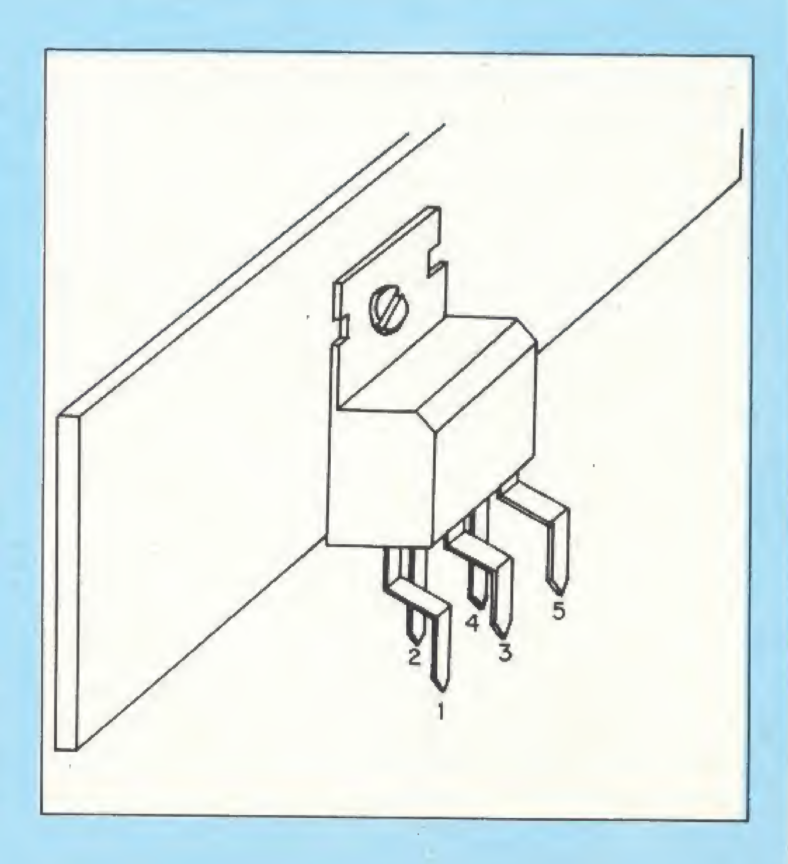




Figura 4. Placa de circuito impreso del amplificador visto por el lado de las pistas de cobre.

Lista de componentes

Para la consola

2 Diodos LED rojos
2 Porta LED
2 Resistencias 680 ohmios 1/4 watio
1 Resistencia, 8 ohmios, 1 watio
3 Conmutadores miniatura de dos canales
1 Interruptor
1 Altavoz de 8 ohmios
1 Portafusibles
8 Conectores y clavijas

Para el amplificador

R1	220	1/4 w	C1	10 μF	3V	Electrolítico
R2	2,2	1/4 w	C2	470 μF	3V	Electrolítico
R3	1	1/4 w	C3	100 μF		
R4	47	1/4 w	C4	1000 μF	10v	Electrolítico
IC1 = TDA 2002			C5	100 μF		
Potenciómetro 10 k log con int.			C6	100 μF		



Alberto Piedra con su montaje.

cacharrear con él, sin demasiado miedo a destruirlo.

Montaje del amplificador

Para montar este circuito utilizaremos la placa de circuito impreso de la figura 4. Una vez fabricada, y efectuados todos los agujeros para introducir los componentes, empezaremos montando las resistencias.

Seguidamente, soldaremos los condensadores, teniendo en cuenta que C1, C2 y C4 son electrolíticos y así pues han de ser soldados en su posición correcta. C3, C5 y C6 carecen de polaridad, así resulta indiferente soldarlos en cualquiera de las dos posiciones. Finalizaremos el montaje soldando el circuito integrado, que no puede ser introducido en la placa más que en una posición, no dando lugar a dudas su alojamiento en la placa.

En el esquema del circuito se encuentra un condensador electrolítico de 100 μF colocado en paralelo con C3, entre masa y alimentación (patilla 5). Este condensador no es necesario, y de hecho no se ha incluido en el circuito impreso, ya que su misión es de filtrar la alimentación que proviene de la fuente de alimentación del Spectrum, que ya posee un condensador electrolítico para filtrar la tensión de alimentación.

En caso de no encontrarse los condensadores electrolíticos al voltaje indicado, también se pueden utilizar, naturalmente, condensadores de la misma capacidad pero a un voltaje mayor, por ejemplo 25 voltios, pero en ningún caso un voltaje inferior al indicado.

Por último, es conveniente atornillar algún tipo de disipador de calor al circuito integrado, para lo cual nos puede valer alguna plaquita de aluminio rectangular de dimensiones 2 × 6 centímetros (ver figura 5).

Alberto Piedra

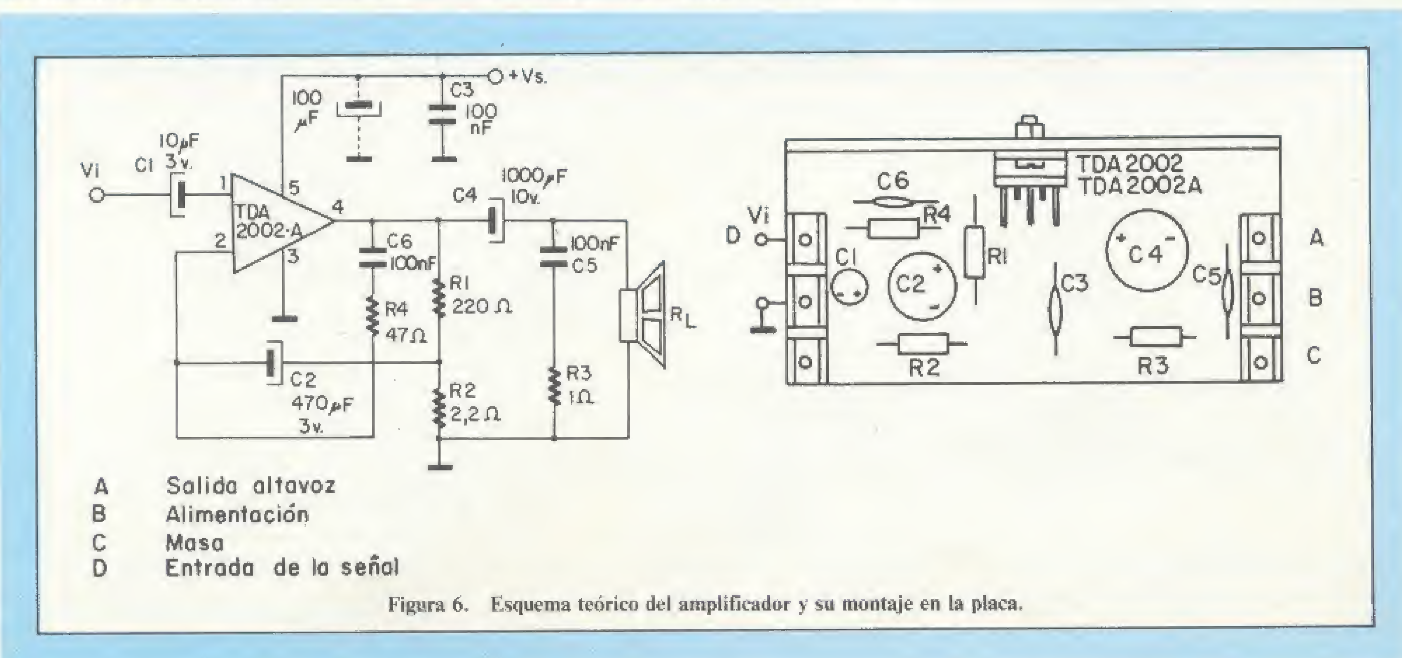
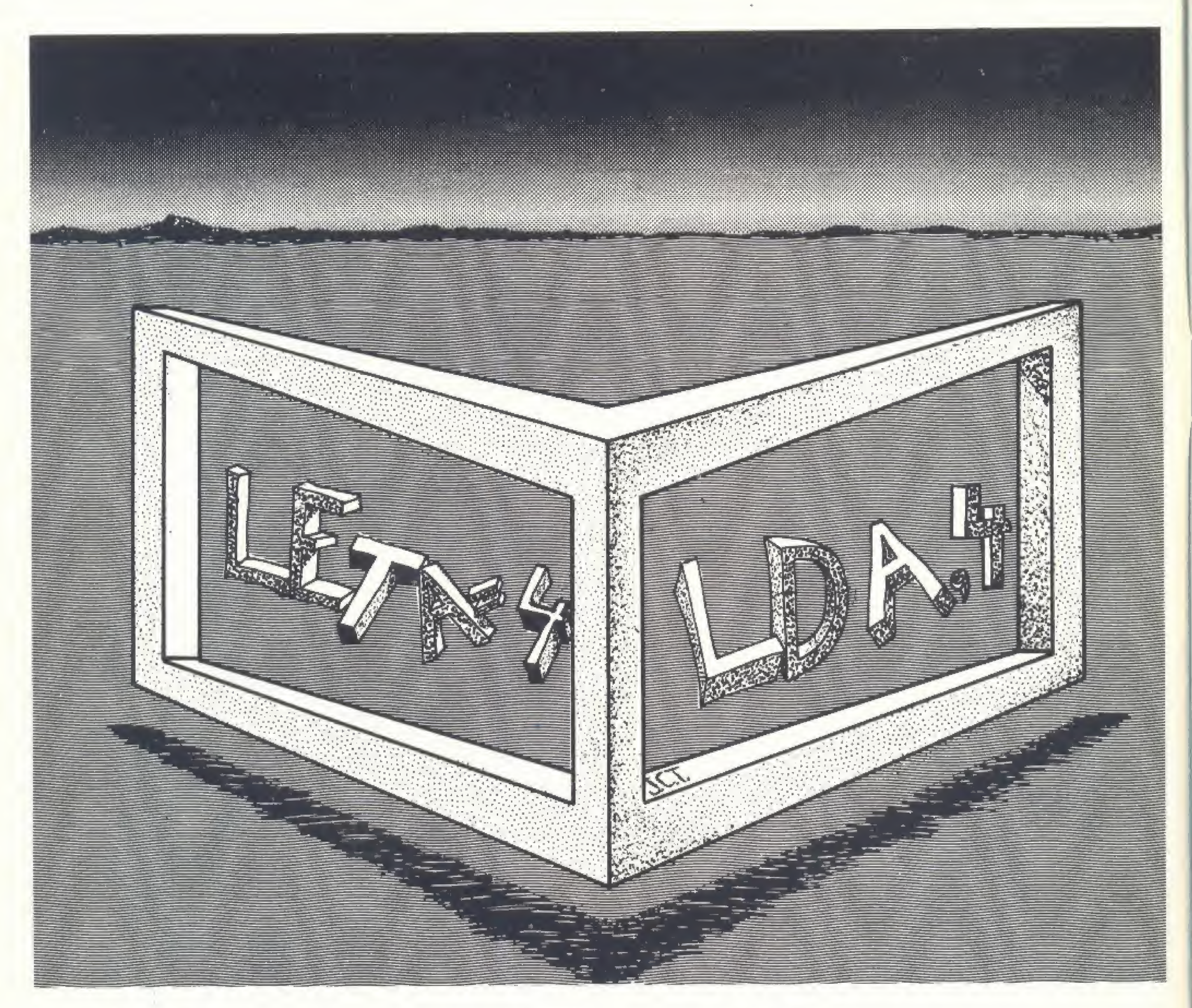


Figura 6. Esquema teórico del amplificador y su montaje en la placa.



investronica
Tomás Bretón, 62
Teléfono (91) 467 82 10
Telex. 23399 IYCO E
28045 MADRID
Comp. 80
08022 BARCELONA
ESPAÑA



A código

Decir que el lenguaje BASIC es lento, no es decir nada nuevo. Olvidarse de él por este motivo y pasar al código máquina es una solución que dista mucho de ser la más eficiente, independientemente del gusto particular de cada uno por este lenguaje. Lo que le proponemos en este artículo es un camino intermedio, potenciar al BASIC con ayuda del código máquina allí donde es más lento. La comparación entre los dos lenguajes, esperamos le sea de utilidad y logre así incrementar la "rapidez" de sus programas.

Para empezar fijemos el problema: Al ordenador le salen telarañas mientras esperas a que te dé unos resultados. Lo primero que hay que hacer es identificar qué parte del programa es la que lo enlentece. Para ello distribuiremos a lo largo del programa diversos puntos de prueba, esto es, mensajes a pantalla, y cronometraremos el tiempo que transcurre entre cada mensaje. Una vez localizado el cuello de botella, procederemos a sustituirlo por una subrutina en lenguaje máquina.

Escribir programas en lenguaje máquina tiene mucho que ver con colocar unos números en unas posiciones de memoria. Para poder hacer esto antes es necesario reservar dichas posiciones de memoria, ya que en principio, toda la memoria está dedicada al BASIC y a los UDG (gráficos definidos por el usuario). En el Spectrum de 48 Kbytes existen 65536 posiciones (del 0 al 65535) que pueden contener números comprendidos entre el 0 y el 255, ambos inclusive.

La reserva de memoria se realiza con la instrucción CLEAR, que fija un tope al área disponible para el programa BASIC. Podremos situar nuestras rutinas, por tanto, por encima de este tope. Situaremos en la primera línea de nuestro programa la instrucción CLEAR 62000 (al final del artículo hay una nota dedicada a los usuarios del Spectrum de 16 Kbytes). Con esto

rápido

habremos logrado fijar el RAMTOP en la posición 62000 y dejando un sitio seguro para nuestro programa en código máquina por encima de esta posición, aunque realicemos un NEW u otro CLEAR. La instrucción CLEAR 62000 nos protegerá 3367 posiciones de memoria, o bytes (65535 menos 62000).

Su programa tomará la forma de una serie de números decimales "pokeados" en posiciones situadas por encima del RAMTOP.

Estos números son utilizados por el ordenador para indicarle al procesador las instrucciones que debe realizar. El intérprete BASIC del Spectrum lo que hace es traducir las líneas BASIC en una serie de estas instrucciones, de manera que el procesador las pueda entender y ejecutar. Los programas en lenguaje máquina son más rápidos que el BASIC porque no necesitan ninguna traducción, son instrucciones ejecutables directamente por el procesador.

El programa 2 utiliza 22 números decimales, situados en las posiciones 62050 al 62071, para generar una serie de tonos musicales cuando se pulsa una tecla. Uno de los programas es más corto gracias a la utilización de las instrucciones DATA, pero los dos realizan la misma función. Una vez que se ejecute el programa BASIC, el programa en código máquina quedará residente en la memoria a partir de la posición 62050 y ya podrá ser borrado al programa en BASIC Para poder ejecutar el código máquina es necesario indicar al ordenador a partir de qué posición debe empezar a ejecutar. Para ello se utiliza la instrucción LET M = USR 62050.

Si esta instrucción está situada dentro de un programa BASIC, después de ejecutarse la rutina en código máquina se seguirá ejecutando el BASIC. En nuestro caso (programa 1) utilizamos algunas líneas para realizar funciones de representación e introducción de datos. Una vez que ya esté ejecutando el programa musical utilice la "B" para acabar.

Se pueden tener diversos programas en código máquina a la vez en memoria con la única, y evidente, condición de que no se solapen. Por ejemplo, se pueden tener subrutinas distintas en las posiciones 62050 al 62071 y 62075 al 6220, y acceder a ellas con las instrucciones LET M = USR 62050 y LET M = USR 62075, respectivamente. Fíjese que dejamos libres entre ellas algunas posiciones que se desperdician, pero que en algún momento pueden llegar a sernos útiles si hacemos alguna modificación en las subrutinas.

Además de las posiciones de memoria el procesador puede acceder a los llamados registros. Los registros son como las posiciones, pero el procesador las suele utilizar para funciones especiales porque son más cómodos y rápidos de utilizar. Los registros se nombran por letras A, B, C, D, E, H y L. El más importante de ellos es el A, llamado también acumulador, ya que es el que contiene la información que está manejando el procesador en cada momento.

Una instrucción típica del ensamblador puede ser LD A,(62001), que significa cargar (LD, load en inglés) el acumulador con el contenido de la posición 62001. Recuerde que el ensamblador es una forma fácil de representar el lenguaje máquina. Utilizamos LD para representar el número decimal 58 que le indica al procesador que debe cargar algo en algún registro.

Si posee algún ensamblador para Spectrum lo único que tiene que saber que LD significa "load" que en castellano quiere decir "carga". Si por el contrario no tiene ensamblador, entonces tendrá que programar directamente con los números. De todas formas, y aunque no tenga ensamblador, le recomendamos que programe utilizando los mnemónicos (abreviaturas) típicas del ensamblador y luego traduzca, a mano, a código máquina.

"Pokeando" los números 58, 49

y 242 en dos posiciones de memoria consecutivas, situadas por encima del RAMTOP, tendrá un pequeño programa en lenguaje máquina que cargará en el acumulador el contenido de la posición 62001. Deberemos añadir también un cuarto número, 201, que se representa por el mnemónico RET (RETurn = volver); para indicar al procesador que hay que volver al BASIC.

Si se fija, se puede extrañar que 58 vaya seguido de dos parámetros: 49 y 242, mientras que LD sólo necesita uno, una dirección. La razón de ello es que para indicar una dirección es necesario un número más grande que 255, que es la mayor cantidad que se puede representar mediante un *byte* (con ocho bits sólo se pueden representar números comprendidos entre 0 y $2^8 -1$). Por lo tanto hacen falta dos *bytes* para representar una dirección. Para calcular qué dos números representan una determinada dirección, puede utilizarse el programa número 3.

En el código máquina pueden encontrarse conceptos equivalentes a los ya encontrados en el BASIC. Por ejemplo: JP (JumP = salto) es equivalente al GOTO del BASIC Una instrucción JP ocupa tres *bytes* de memoria. El primero es un 195 (decimal), y los siguientes los dos números que representan a la dirección a donde se va a efectuar el salto. Por el contrario, el código máquina carece de todas las protecciones que se pueden encontrar en el BASIC. Por ejemplo, en BASIC si se ejecuta una instrucción del tipo 10 GOTO 10 el ordenador entra en un bucle sin fin, pero que puede ser interrumpido pulsando las teclas CAPS y SPACE. En código máquina jamás se puede interrumpir un proceso, a no ser que el propio programa tenga explícita una salida, examinando el teclado, por ejemplo. Una buena idea en un programa máquina es comprobar el teclado, y si ha sido pulsado, volver al BASIC. Esto nos evitará dejar la máquina colgada en un bucle infinito, y sin salida posible.

Además, cuando se trabaja en código máquina, no existe comprobación sintáctica, por tanto, to-

PROGRAMA 1

```
   1 CLEAR 62000
   2 FOR i=62050 TO 62060
   5 READ x: POKE i,x: NEXT i
  10 DATA 205,142,2,40,2,30,254,
6,0,75,201
  12 LET m=USR 62050
  14 PRINT AT 0,0;" "
  15 GO TO 12
```

LD A, (dir.)	Carga el acumulador con el contenido de la posición 62004	62050	58	58
		62051	Y	52
		62052	Y	242
CPL	Comprueba si el contenido del acumulador es 1: de ser así...	62053	254	254
		62054	X	1
JPZ, (dir.)	... salta a la posición 62073	62055	202	202
		62056	Y	121
		62057	Y	242
CP2	Comprueba si el contenido del acumulador es 2: de ser así...	62058	254	254
		62059	X	2
JPZ, (dir.)	... salta a la posición 62087	62060	202	202
		62061	Y	135
		62062	Y	242
CP3	Comprueba si el contenido del acumulador es 3: de ser así...	62063	254	254
		62064	X	3
JPZ, (dir.)	... salta a la posición 62101	62065	202	202
		62066	Y	149
		62067	Y	242
CP4	Comprueba si el contenido del acumulador es 4: de ser así...	62068	254	254
		62069	X	4
JPZ, (dir.)	... salta a la posición 62015	62070	202	202
		62071	Y	163
		62072	Y	242
LD A, (dir.)	Carga el acumulador con el contenido de la posición 62001	62073	58	58
		62074	Y	49
		62075	Y	242
DEC A	Decrementa el acumulador una unidad	62076	61	61
LD, (dir.) A	Carga la posición 62005 con el contenido del acumulador	62077	50	50
		62078	Y	53
		62079	Y	242
LD A, (dir.)	Carga el acumulador con el contenido de la posición 62002	62080	58	58
		62081	Y	58
		62082	Y	242
LD, (dir.) A	Carga la posición 62006 con el contenido del acumulador	62083	50	50
		62084	Y	54
		62085	Y	242
RET	Vuelta al BASIC	62086	201	201
LD A, (dir.)	Carga el acumulador con el contenido de la posición 62001	62087	58	58
		62088	Y	49
		62089	Y	242

PROGRAMA 2. METODO 1

```
 1 CLEAR 62000
 2 FOR I=62050 TO 62071
 3 READ X: POKE I,X: NEXT I
 4 DATA 205,142,2,123,254
 5 DATA 0,202,119,242,103
 6 DATA 111,22,0,30,0
 7 DATA 205,181,3,195,98
 8 DATA 242,201
12 LET M=USR 62050
```

PROGRAMA 2. METODO 2

```
  1 CLEAR 62000: REM fija RAMTO
P
  2 POKE 62050,205        14 POKE 62062,0
  3 POKE 62051,142        15 POKE 62063,30
  4 POKE 62052,2          16 POKE 62064,0
  5 POKE 62053,123        17 POKE 62065,205
  6 POKE 62054,254        18 POKE 62066,181
  7 POKE 62055,0          19 POKE 62067,3
  8 POKE 62056,202        20 POKE 62068,195
  9 POKE 62057,119        21 POKE 62069,98
 10 POKE 62058,242        22 POKE 62070,242
 11 POKE 62059,103        23 POKE 62071,201
 12 POKE 62060,111       100 LET M=USR 62050
 13 POKE 62061,22
```

INC A	Incrementa el acumulador en una unidad	62090	60	60
LD, (dir.) A	Carga la posición 62005 con el contenido del acumulador	62091 62092 62093	50 Y Y	50 53 242
LD A, (dir.)	Carga el acumulador con el contenido de la posición 62002	62094 62095 62096	58 Y Y	58 50 242
LD, (dir.) A	Carga la posición 62006 con el contenido del acumulador	62097 62098 62099	50 Y Y	50 54 242
RET	Vuelta al BASIC	62100	201	201
LD A (dir.)	Carga el acumulador con el contenido de la posición 62002	62101 62102 62103	58 Y Y	58 50 242
DEC A	Decrementa el acumulador en una unidad	62104	61	61
LD, (dir.) A	Carga la posición 62006 con el contenido del acumulador	62105 62106 62107	50 Y Y	50 54 242
LD A, (dir.)	Carga el acumulador con el contenido de la posición 62001	62108 62109 62110	58 Y Y	58 49 242
LD, (dir.)	Carga la posición 62005 con el contenido del acumulador	62111 62112 62113	50 Y Y	50 53 242
RET	Vuelta al BASIC	62114	201	201
LD A, (dir.)	Carga el acumulador con el contenido de la posición 62002	62115 62116 62117	58 Y Y	58 50 242
INC A	Incrementa el contenido del acumulador en una unidad	62118	60	60
LD, (dir.) A	Carga la posición 62006 con el contenido del acumulador	62119 62120 62121	50 Y Y	5050 54 242
LD A, (dir.)	Carga el acumulador con el contenido de la posición 62001	62122 62123 62124	58 Y Y	58 49 242
LD, (dir.) A	Carga la posición 62005 con el contenido del acumulador	62125 62126 62127	50 Y Y	50 53 242
RET	Vuelta al BASIC	62128	201	201

dos los errores son tragados y ejecutados por la máquina. Lo más probable es que, si se comete un error, la máquina se "cuelgue", y que la única forma de salir sea desenchufando y volviendo a enchufar.

Algunas veces parecerá que la máquina produce mensajes de error. Estos falsos mensajes de error se producen cuando antes de volver al BASIC se han alterado posiciones por debajo del RAMTOP, dañando así el programa BASIC.

Todo lo anteriormente citado hace a los programas en código máquina muy difíciles de depurar. Un ensamblador puede resultar muy útil (los mnemónicos son más fáciles de leer) y además algunos ensambladores proporcionan herramientas de depuración. Sin un ensamblador la realización de un programa mínimamente complejo puede resultar penosa.

Cuando diseñe un programa en código máquina hágalo de forma que sea fácil de corregir y depurar. Es también buena idea diseñar un número de programas pequeños, probados por separado, y luego unirlos, que diseñar un único programa grande, más difícil de comprobar. Por ejemplo, si se quiere hacer un programa para dibujar figuras coloreadas en la pantalla, resultará más fácil dividir el programa en dos subprogramas, de la siguiente manera: Un programa para dibujar figuras y otro programa para colorear figuras. Siempre serán más fáciles de corregir por separado.

Para escribir un programa en có-

digo máquina conviene seguir un procedimiento formal fijo. Lo primero que hace falta es una lista de los códigos decimales de las instrucciones (apéndice A del manual del Spectrum). También es necesario saber qué funciones realiza cada instrucción. No tenemos sitio aquí para explicar casi 700 instrucciones, que son todas las que tiene el procesador Z80, así que nos limitaremos a explicar una selección de las más utilizadas, y que nos bastarán para realizar un ejemplo de cómo transformar un programa BASIC en lenguaje máquina.

Este que sigue, es el programa a traducir:

65 GOSUB 90+H: REM H PUEDE VALER 1,2,3 ó 4
" "... (Resto del programa)
91 LET YF=YI-1:LET XF=XI : RETURN
92 LET YF=YI+1:LET XF=XI : RETURN
93 LET XF=XI-1:LET YF=YI : RETURN
94 LET XF=XI+1:LET YF=YI : RETURN

Estas líneas pueden pertenecer a cualquier juego. Los valores de H significan un desplazamiento hacia arriba, abajo, derecha o izquierda. XI e YI son las coordenadas iniciales (antes del desplazamiento), y XF e YF son las finales. Damos por supuesto que, por programa, H sólo puede tomar los valores arriba indicados.

Supongamos que el programa fue escrito, desarrollado y depurado en BASIC y que cuando se ejecutó se comprobó con horror que resultaba muy lento. La solución consiste en ir sustituyendo segmentos del programa BASIC por rutinas en código máquina hasta conseguir la velocidad adecuada. No es necesario traducir todo el programa.

El resultado es un programa formado en gran parte por una estructura BASIC que va llamando a pequeñas rutinas en código máquina, encargadas de realizar los trabajos que hubieran resultado muy lentos en BASIC.

Antes de describir las etapas que nos van a llevar a convertir en má-

PROGRAMA 3

```
  10 INPUT "Escribe la direccion
 ";DIR
  20 CLS : PRINT "La direccion "
;DIR
  30 PRINT "se representa por lo
s numeros"
  40 PRINT DIR-INT (DIR/256)*256
;" Y ";INT (DIR/256)
  50 PAUSE 0: GO TO 10
```

PROGRAMA 5

```
   1 CLEAR 62000
   2 FOR I=62050 TO 62128
   3 READ X: POKE I,X: NEXT I
   4 DATA 58,52,242,254,1,202,12
1,242,254,2
   5 DATA 202,135,242,254,3,202,
149,242,254,4
   6 DATA 202,163,242,58,49,242,
61,50,53,242
   7 DATA 58,50,242,50,54,242,20
1,58,49,242
   8 DATA 60,50,53,242,58,50,242
,50,54,242
   9 DATA 201,58,50,242,61,50,54
,242,58,49
  10 DATA 242,50,53,242,201,58,5
0,242,60,50
  11 DATA 54,242,58,49,242,50,53
,242,201
  12 REM ** CODIGO MAQUINA **
  13 REM **       EN       **
  14 REM **     MEMORIA    **
  15 REM
  16 REM --LINEAS DE PRUEBA--
  17 REM
  39 FOR H=1 TO 4
  40 POKE 62004,H
  50 POKE 62001,4: REM Var. YI
  60 POKE 62002,4: REM Var.XI
  70 LET M=USR 62050
  80 PRINT "------------------"
  90 PRINT "Direc.62004 (antes H
)= ";PEEK 62004
 100 PRINT "Direc. 62005 (antes
YF)= ";PEEK 62005
 110 PRINT "Direc. 62006 (antes
XF)= ";PEEK 62006
 120 NEXT H
```

```
Direc. 62004 (antes H)= 1
Direc. 62005 (antes YF)=3
Direc. 62006 (antes XF)=4
-------------------------
Direc. 62004 (antes H)= 2
Direc. 62005 (antes YF)=5
Direc. 62006 (antes XF)=4
-------------------------
Direc. 62004 (antes H)= 3
Direc. 62005 (antes YF)=4
Direc. 62006 (antes XF)=3
-------------------------
Direc. 62004 (antes H)= 4
Direc. 62005 (antes YF)=4
Direc. 62006 (antes XF)=5
```

PROGRAMA 6

```
 1 CLEAR 62000
 2 FOR i=62050 TO 62060
 5 READ x: POKE i,x: NEXT i
10 DATA 205,142,2,40,2,30,254,
6,0,75,201
12 LET m=USR 62050
14 PRINT AT 0,0;m;" "
15 GO TO 12
```

Código máquina	Decimal	Explicación
LD A (dir)	58 N N	Carga el acumulador con el contenido de una posición de memoria, como la 62001 Ejemplo: 62050 [58] 62051 [49] 62052 [242] 62053 [ ] LD A(62001) Los dos N definen la posición 62001. Deben tomar los valores 49 y 242 en este ejemplo. Ver programa 3 para más aclaraciones
JP dir	195 N N	Salto a la posición definida por las dos N. Ejemplo: 62050 [195] 62051 [104] 62052 [242] 62053 [ ] 62054 [ ] 62055 [ ] 62056 [ ] 62057 [ ] JP 62056 La posición, en este caso 62056, viene definida por los números 104 y 242
RET	201	Vuelta al Basic 100 LET M=USR 62050 101 REM 62050 [ ] 62051 [ ] 62069 [ ] 62070 [201] RET Instrucciones en código máquina
CALL dir y su compañero	205 N N	En este caso RET no actúa por separado, sino como compañero del CALL, por tanto no producirá la vuelta al BASIC. CALL es equivalente a la instrucción BASIC GOSUB Ejemplo: 62050 [205] 62051 [104] 62052 [242] 62053 [ ] 62054 [ ] 62055 [ ] 62056 [ ] 62057 [ ] 62058 [ ] 62059 [201] CALL 62056 Vuelta al punto desde el que se efectuó el CALL, no al BASIC. Saltamos desde la posición 62052, y volvemos a la 62063, así que no se vuelve exactamente a la posición de llamada, sino a la siguiente

NOTA: Los recuadros en blanco contienen otras instrucciones.

REGLAS:

a) Si encuentra un CALL dir, entonces el RET es su compañero.

b) Si no hay CALL, entonces el retorno se realiza al BASIC.

Este es un ejemplo de la utilización de CALL y su compañero RET.

quina el programa de arriba, sustituyamos las variables por posiciones de memoria. A partir de ahora sustituiremos YI por la posición 62001. Para ver su valor utilizaremos la función PEEK y para cambiarlo POKE. Por ejemplo, si en una línea de programa tenemos la instrucción PRINT YI, la sustituiremos por PRINT PEEK 62001. Si encontramos LET A=YI lo sustituiremos por LET A = PEEK 62001. Si lo que encontramos es LET YI = 4, entonces lo cambiaremos por POKE 62001,4.

Podemos sustituir YI por una dirección porque sabemos que tomará valores entre 0 y 255, ya que es el valor de una coordenada de pantalla. Haremos lo mismo con las variables H, XI, YF y XF y las posiciones 62004, 62002, 62005 y 62006, respectivamente. Aunque todavía estemos en BASIC (PEEK y POKE pertenecen al BASIC), ya hemos empezado la transformación. Este cambio nos permite compartir valores entre los dos lenguajes.

La figura 4 indica los pasos seguidos para transformar nuestro programa. Las dos primeras columnas contienen las instrucciones en ensamblador y algunos comentarios; la tercera indica el número de las posiciones de memoria; la cuarta, el código operativo de la instrucción, así como los parámetros requeridos por cada instrucción; en la quinta, están los valores definitivos que toman las posiciones de memoria. Todos los valores están dados en decimal.

En el programa 5 vemos el programa BASIC utilizado para introducir en memoria el código máquina y para probarlo. También hay unas copias de las pantallas resultantes de la ejecución. Una vez que el programa haya pasado todos los controles puede añadirse al programa principal, simplemente incluyendo unas líneas que "pokeen" los números en las posiciones adecuadas por encima del RAMTOP. Una vez hecho esto, pueden borrarse las líneas BASIC 65, 91, 92, 93 y 94 y sustituirlas por una línea 64 LET M = USR 62050.

Recuerde que ahora las antiguas variables no sirven para nada, así

Código máquina	Decimal	Explicación
LD(dir)A	50 N N	Guarda el contenido del acumulador en la posición definida por las dos N. Ejemplo: 62053 [50] / 62054 [50] / 62055 [242] } LD(62002)A / 62056 [ ] / 62057 [ ] He aquí un programa en código máquina completo que transfiere el contenido de la posición 62001 al acumulador, y de aquí a la posición 62002. Después vuelve al BASIC. **Listado** 10 CLEAR 62000: REM fija ram top 20 FOR i=62050 TO 62056 30 READ x: POKE i,x: NEXT i 40 DATA 58,49,242,50,50,242,20,1 50 LET m=USR 62050 60 STOP 62050 [58] / 62051 [49] / 62052 [242] } LD A(62001) 62053 [50] / 62054 [50] / 62055 [242] } LD(62002)A 62056 [201] RET 49 y 242 definen el 62001 50 y 242 definen el 62002 Antes de ejecutar el programa en código máquina. 62001 [81] 62002 [79] Después de ejecutar el programa. 62001 [81] 62002 [81]
Instrucciones similares a la anterior		
LD B,n	6	
LD C,n	N 14	
LD C,n	N 22	
LD D,n	N 30	
LD E,n	N 38	
LD H,n	N 46	
LD L,n	N	
LD A,N	62 n	Carga el acumulador con el número "N". "N" debe estar comprendido entre 0 y 255. Ejemplo: 62050 [62] / 62051 [3] } LD A,3
DEC A	61	Decrementa un contenido del acumulador en una unidad. NOTA: 0 menos 1 produce 255.
INC A	60	Incrementa el contenido del acumulador en una unidad. NOTA: 255 más 1 produce 0.

que hay que eliminarlas por completo. En caso contrario podrían darnos problemas. Para cerciorarnos de que las hemos eliminado, después de ejecutar el programa ejecutaremos la instrucción PRINT YI, por ejemplo. Si el ordenador da el mensaje *Variable not found* sabremos que nuestro programa no utiliza para nada la variable YI.

Ahora ya tenemos tres programas. El programa BASIC, el programa BASIC que "pokea" el programa en código máquina y el programa en código máquina en sí. Guarde una copia del programa "pokeador" con fines de documentación. Salve el programa en código máquina utilizando una instrucción SAVE "nombre" CODE m,n. Donde "m" es la dirección de comienzo de la rutina (62050 en nuestro caso) y "n" la longitud del programa (78). Es decir: SAVE "nombre" CODE 62050,78. Para volver a cargar la rutina máquina escriba LOAD " " CODE.

Como puede ver en el programa 4, traducir algunas instrucciones BASIC es relativamente sencillo y no necesita tener un conocimiento exacto del conjunto completo de instrucciones del Z80. Aunque pueda parecer en principio trivial, es una buena forma de empezar a adentrarse en los misterios del código máquina. Comience traduciendo los condicionales y los bucles que son las instrucciones que, normalmente, enlentecen más un programa. Cuando tenga alguna soltura puede empezar a encargarse de segmentos de programa más extensos o complicados, como pueden ser la entrada y la salida. Una línea que imprima algo es mucho más fácil de programar en BASIC que en código máquina, pero es también mucho más lenta de ejecutar.

Por último, una nota aclaratoria para los usuarios del Spectrum 16 Kbytes: existen 32768 posiciones de memoria, así que escriba CLEAR 30000 en vez de 62000. Utilice las posiciones 30001 al 30049 para los datos, y sitúe el programa en código máquina a partir de la posición 30050.

Spectrum puede con todos.

¿Quién nos gana en gama? Estamos por asegurar que ninguno. No olvides que tenemos un Spectrum para cada exigencia: dos capacidades diferentes (16K y 48K) y tres modelos con dos tipos de teclado (doméstico y profesional).

¿Quién nos gana en programas? Spectrum cuenta con más de 5.000 títulos publicados a nivel internacional, cien de ellos están traducidos al castellano.

Naturalmente estos crecen casi de forma constante. Una buena muestra es el voluminoso catálogo de software que puedes solicitar a tu distribuidor de confianza.

¿Quién nos gana en periféricos? Ya son más de 50 los periféricos creados especialmente para el Spectrum, pero no creas que eso termina ahí. Es muy raro el día que no aparece en el mercado una novedad. Así tu Spectrum guardará para tí el mismo interés del primer día.

¿Quién nos supera en número? Otro factor a tener en cuenta: te diremos que ya son más de tres millones los microordenadores Sinclair vendidos en todo el mundo (y más de 100.000 Spectrum vendidos en España) ¿no te parece esto una buena razón para confiar en tu Spectrum?.

Decídete; este año tener un Spectrum es todo un regalo.

Los concesionarios INVESTRONICA tienen para tí un montón de novedades.

* **De venta en la Red de Concesionarios INVESTRONICA.**



SINCLAIR RESEARCH LIMITED hace constar que no está en condiciones de garantizar el origen y calidad de aquellos productos que no hayan sido comercializados en España a través de su distribuidor exclusivo INVESTRONICA, S. A.

Código máquina	Decimal	Explicación
ADD A,n	198 n	Suma al acumulador el número decimal "N", comprendido entre 0 y 255. Ejemplo: 62050 [198] 62051 [24] } ADD A,24 Si A=6 antes, entonces después A=30
SUB n	214 n	Resta del acumulador el número decimal "N", comprendido entre 0 y 255. Ejemplo: 62050 [214] 62051 [13] } SUB 13
ADD A,B	128	Suma el contenido del registro B al A (acumulador), dejando el resultado en el A. Ejemplo: A [7] B [2] / A [9] B [2]

Instrucciones similares a la anterior
ADD A,C 129
ADD A,D 130
ADD A,E 131
ADDA,H 132
ADD A,L 133

LD B,A	71	Carga en el registro B el contenido del A. Para conocer más instrucciones similares a ésta, consultar la página 185 del manual del Spectrum. Por ejemplo, al final de la página se puede leer: 125 70 ld a,l bit 7,l LD A,L (o ld a,l) tienen el código operativo 125. 7D es el número equivalente en hexadecimal.
CP n seguido de	254 n	Este par de instrucciones significan comparar el contenido del acumulador con el número "n", y si son iguales saltar a "dir".
JP Z(dir)	202 N N	"dir" está definido por las dos "N". "n" es un número entre 0 y 255. Ejemplo:

62050 [58]
62051 [49]
62052 [242]
62053 [254]
62054 [3]
62055 [202]
62056 [1]
62057 [243]
⋮
62209 [ ]
62210 [ ]

He aquí un programa que salta a la posición 62209 si en la posición 62001 hay un 3.

LD A (62001)
Cpn
seguido de JP Z, 62209
Saltar si el contenido del acumulador es un 3.

Código máquina	Decimal	Explicación
Varias juntas	205 142 2 40 2 30 254 6 0 75	Se mira el teclado, quedando el registro C con los siguientes valores: — Si no se ha pulsado ninguna tecla, C contiene 255. — Si se ha pulsado alguna tecla entonces queda el valor especificado en la tabla siguiente

Tecla:	A	B	C	D	E
Valor:	38	0	15	22	21
Tecla:	F	G	H	I	J
Valor:	14	6	1	18	9
Tecla:	K	L	M	N	O
Valor:	17	25	16	8	26
Tecla:	P	Q	R	S	T
Valor:	34	37	13	30	5
Tecla:	U	V	W	X	Y
Valor:	10	7	29	23	2
Tecla:	Z	1	2	3	4
Valor:	31	36	28	20	12
Tecla:	5	6	7	8	9
Valor:	4	3	11	19	27
Tecla:	0	CAPS	SHIFT		
Valor:	35	39	24		

— Si se pulsa más de una tecla a la vez, C contendrá el valor 254, salvo que una de las teclas sea SHIFT en cuyo caso quedará el valor correspondiente a la otra tecla. Por ejemplo, 31 de haberse pulsado la Z.

Ejemplo:
62050 [205]
62051 [142]
62052 [2]
62053 [40]
62054 [2]
62055 [30]
62056 [254]
62056 [6]
62057 [0]
62058 [75]
62059 [ ]

Leer el teclado, y situar un valor en el registro C.

Finalmente, el programa 6 es un programa BASIC que accede al teclado. Cuando esté ejecutando el programa pulse cualquier tecla, y el valor del registro C aparecerá en pantalla.

```
1 CLEAR 62000
2 FOR i=62050 TO 62060
5 READ x: POKE i,x: NEXT i
10 DATA 250,142,2,40,2,30,254,6,0,75,201
12 LET m=USR 62050
14 PRINT AT 0,0;m; " "
15 GO TO 12
```

SUSCRIBASE A

Todospectrum

(12 NUMEROS)

TARIFA DE PRECIOS DE SUSCRIPCION

	CORREO ORDINARIO		CORREO CERTIFICADO		CORREO AEREO		CORREO AEREO-CERTIF.	
	PTAS.	$	PTAS.	$	PTAS.	$	PTAS.	$
ESPAÑA	3.000	21	3.273	23	3.055	22	3.333	24
EUROPA, MARRUECOS, TUNEZ, TURQUIA, ARGELIA Y CHIPRE	3.456	25	4.272	31	3.600	26	4.418	31
COSTA RICA, CUBA, CHILE, PARAGUAY Y REP. DOMINICANA	3.396	24	4.212	30	4.164	30	4.980	36
GIBRALTAR Y PORTUGAL	3.264	23	4.080	29	3.149	22	3.965	28
FILIPINAS	3.264	23	3.540	25	3.775	27	4.050	29
RESTO DEL MUNDO	3.456	25	4.272	31	4.224	30	5.040	36

CUPON DE PEDIDO

Recorte y envie este cupón a: **Todospectrum** EDISA. Lopez de Hoyos, 141 - 28002 - MADRID

El importe lo abonaré: POR CHEQUE ☐ CONTRA REEMBOLSO ☐
CON TARJETA DE CREDITO ☐ American Express ☐ Visa ☐ Interbank ☐

Número de mi Tarjeta: ☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐ Fecha de caducidad: ____________

NOMBRE ____

DIRECCION ____

CIUDAD ______________________ D.P. ____ PROVINCIA ______________



MAYORES PRESTACIONES EN MICRODRIVE

Aun desconocidos por la mayoría, los programas de contabilidad para el Spectrum van creciendo tanto en cantidad como en calidad. Una de las aplicaciones clásicas por las que entraba un ordenador en las empresas, le permite ahora al Spectrum competir ya sea en el terreno personal o profesional. **World-Micro** lo ha hecho posible.

Uno de los mayores problemas de todo programa de contabilidad reside en la limitación del número de asientos o apuntes, y en el número de cuentas a utilizar en los distintos grupos y subgrupos. Para lograr resultados aceptables ha de poderse trabajar con el mayor número posible en ambos casos, pero muy especialmente por lo que se refiere a los apuntes contables, conocido por el diario. Al tener un número de datos elevado surge el problema de su manipulación para acceder a los datos desde memorias externas (léase *microdrives*) de la forma más rápida y eficiente posible. A esto ha de sumarse el requisito de la ordenación por fechas para obtener la información actualizada en todo momento y resolver los típicos problemas de la factura que aparece «dos meses después». Los famosos contra-asientos pasan al baúl del olvido, gracias a elaboradas técnicas de programación que pueden dar verdaderos quebraderos de cabeza cuando dejan de funcionar.

Pensando en todos estos problemas elegimos el programa de contabilidad **World Micro** por ser uno de los que ofrecen un mayor número de apuntes y cuentas. Esperamos que el análisis detallado de las posibilidades de este programa le proporcione mayor información de la que puede esperar de su pro-

grama de contabilidad y elija acorde a sus necesidades.

Si se decide por este programa, puede elegir entre utilizar uno o dos *microdrives.* Esta segunda opción es más cómoda, pero más cara ya que necesitará de las dos unidades. Si estaba pensando en utilizar un *cassette,* mejor piénselo dos veces. Peso si estaba pensando en comprarlo y hacer copias, mejor olvídelo. El programa necesita una placa de protección que viene con el cartucho y que ha sido diseñada especialmente como sistema «antipiratas». Esta placa se conecta en el port de expansión. Una vez que ya dispone del ordenador, *interface* 1, unidad(es) de *microdrive,* programa e interface ya puede empezar a trabajar. Necesitará, además, una copia impresa de su trabajo, es decir una impresora y aquí siempre tendrá problemas independientemente del programa que elija. Este programa está preparado para la impresora **Newprint Admate DP80** conectada al *interface* 1. Para trabajar con otras impresoras cerciórese primero de que funcionen y pague después.

Como decíamos, para trabajar con el programa son necesarios dos cartuchos de *microdrive.* En uno se encuentran todos los programas a los que se accede según se vayan necesitando para mantener la RAM lo menos ocupada en cuanto a programas se refiere, y en el otro están los archivos de datos. Lógicamente, con dos *microdrives* puede trabajar más rápidamente. Y como la velocidad es un factor muy importante, trabaja generalmente en memoria, motivo por el que los programas han de ser cortos. La capacidad de los archivos es de 256 cuentas que pueden desarrollarse en dos niveles, y 1000 apuntes. Para lograr una rapidez aceptable, imposible de conseguir si se grabasen los datos cada vez que se finaliza la introducción de un apunte, se van guardando los datos en memoria (RAM) y se graban en bloque cuando se indica el fin de trabajo o cuando se superan los 500 apuntes. (Esto es debido a que los 1000 apuntes se guardan en dos bloques de 500 cada uno, no pudiéndose disponer de los 1000 simultáneamente por problemas de memoria). Aquí tenemos una solución al problema de los tiempos de acceso en *microdrive,* pero que nos generará problemas mayores si la estabilidad de la corriente no es buena o si por azar se produce alguna interferencia. Quedarse «colgado» puede significar perder todo el trabajo de un día. Solución: no esperar al final del día para la grabación. Se puede indicar fin de trabajo y reanudar posteriormente la introducción de datos.

Conocida la estructura básica del programa, pasemos a algo más operativo. Introduciendo el cartucho de programas, habiendo conectado previamente la placa de protección, se accede al programa directamente pulsando RUN y ENTER. De esta forma aparecerá el menú de opciones:

1. Gestión del plan contable.
2. Gestión del fichero de apuntes.
3. Balances y extractos.
4. Actualización.
5. Gestión del cartucho.
6. Fin de sesión.

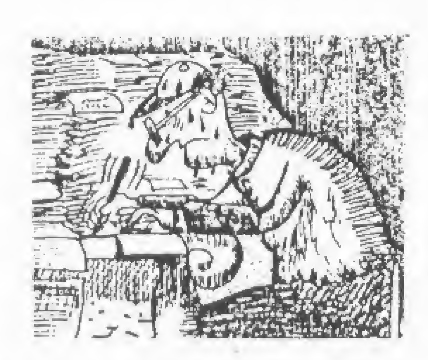
Para empezar a trabajar hay que preparar el cartucho. De esto se encarga la opción cinco, gestión

```
                              LISTADO PLAN CONTABLE                    PAGINA 1
****************************************************************************************
NS CUENTA  *TITULO                            *SALDO I. *   DEBE  *  HABER  *PRESUPUESTO
10
 10000      CAPITAL
 10100      CAPITAL SOCIAL
 10200      FONDO SOCIAL
            CAPITAL
11
 11100      RESERVAS
 11200      PLUSVALIA REVALOR. ACTIVO
 11300      CUENTA DE REGULARIZACION
 11400      RESERVAS LEGALES
 11500      RESERVAS ESPECIALES
 11600      RESERVAS ESTATUARIAS
            RESERVAS VOLUNTARIAS
12
 12000      PREVISIONES
            PREVISIONES PARA RIESGOS
13
 13000      RESULTAD.PENDIEN.APLICAC.
 13100      REMANENTE
 13110      AMORTIZABLES FISCALMENTE
            NO AMORTIZABLES FISCALME.
14
 14000      SUBVENCIONES EN CAPITAL
            SUBVENCIONES OFICIALES
15
 15000      EMPRESTITOS
            OBLIGAC. Y BONOS SIMPLES
17
 17000      PREST.REC.OTROS D.E.F.G.
 17100      PREST.REC.OTROS D.E.F.G.
            PRESTAMOS A PLAZO MEDIO
18
 18000      FIANZAS Y DEPOSIT.RECIBI.
            FIANZAS Y DEPOSIT.RECIBI.
19
 19000      SITUAC.TRANSITOR.FINANCI.
 19200      ACCION.CAPIT.SIN DESEMBO.
            SOCIOS PARTE NO DESEMBOL.
20
 20000      INMOVILIZADO MATERIAL
 20200      TERRENOS Y BIENES NATURA.
 20300      EDIFICIOS Y OTRAS CONSTR.
 20400      MAQUINARIA E INSTALACION
 20500      ELEMENTOS DE TRANSPORTE
 20600      MOBILIARIO Y ENSERES
 20900      EQUIP.PARA PROCES.INFORM.
            INSTALACIONES COMPLEJAS
21
 21100      INMOVILIZADO INMATERIAL
            PROPIEDAD INDUSTRIAL
```

Listado del plan contable. Nivel de desagregación a 2 y 5 dígitos, con posibilidad de fijar presupuestos.



Una protección *soft* impide la piratería del programa grabado en *microdrive.*



del cartucho, y de algunas operaciones más: formatea y crea los archivos, copia de seguridad y cierre del ejercicio. El primer paso es, pues, formatear y crear los archivos. El siguiente paso, igualmente obligatorio, es crear el plan de cuentas. Para introducir los apuntes hay que referenciarlos a las cuentas ya existentes.

La introducción de datos (cuentas y apuntes) se realiza mediante el procedimiento de máscaras. Las teclas del cursor se emplean para el movimiento horizontal dentro del campo y el vertical para pasar al campo anterior o posterior según la tecla que se presione. Pulsando EDIT se reproduce el campo con el contenido del último dato introducido para este campo. Con TRUE VIDEO se limpia el campo en que se encuentra el cursor. Con DELETE se elimina el carácter en la posición donde se encuentra el cursor. Esta manipulación de los datos es común para todos los campos y una vez que se domina resulta muy práctico para acceder rápidamente a los datos que se desean y producir, en su caso, los cambios que sean necesarios.

Además, una pequeña rutina permite aprovechar la información ya escrita en el campo; así, por ejemplo, si se tiene la cantidad 20.000, para introducir 35.000 basta con pulsar 35 y ENTER, ya que la información de los nuevos campos se escribe sobre el dato anterior.

El plan de cuentas

Al margen de la introducción de los datos de cada cuenta, se ofrece una información útil para el control: número de orden de la cuenta, cuenta anterior, actual y posterior. Una línea final indica si se está insertando una cuenta adicional o cambios en una cuenta ya existente. También va informando de las distintas posibilidades de actuación en cada campo (ñ en campos alfanuméricos, signos negativos en campos numéricos, etc.). Las cuentas quedan automáticamente ordenadas siguiendo el plan contable (10, 100000, 20, 200000, etc.).

“A mayores prestaciones... menor número de asientos y cuentas.”

Es fácil la corrección, ya que al querer introducir una cuenta, si ésta existe aparecen sus datos en los respectivos campos para pasar a su corrección. Pero curiosamente no existe la opción de anulación de las cuentas. Una última característica a destacar es la posibilidad de visualizar, con este mismo procedimiento, las distintas cuentas, no produciéndose la grabación mientras no se introduzcan o modifiquen algunas cuentas.

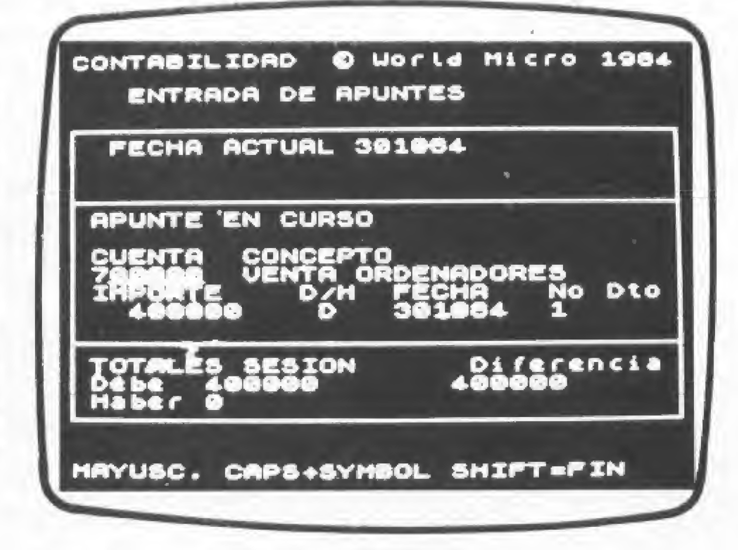
Cuentas	256
Dígitos por cuenta	6
Caracteres por título de cuenta	25
Dígitos en saldo inicial, acumulado debe y haber ...	8
Dígitos para el presupuesto	8

Gestión fichero de apuntes

La entrada de datos se hace, como decíamos, por el procedimiento de máscaras. Primero se indica la fecha de trabajo. Después vienen los datos de los apuntes: cuenta, concepto, importe, debe-



Entrada de los datos de las cuentas.



Entrada de apuntes, con indicativo de totales por sesión y descuadre.

/haber y fecha. Aunque la fecha parezca una redundancia no lo es. El campo de la segunda fecha siempre contiene la introducida en el primero, pudiéndose cambiar para asientos correspondientes a otro día. Es normal realizar varios apuntes con un solo día, incluso cuando corresponden a días distintos, por lo que este pequeño truco es interesante.

Al margen de estos datos, aparece en pantalla más información referente al número de apunte, los totales de DEBE y HABER «por sesión» y la diferencia entre estos conceptos. A ojos de contable, este «por sesión» resultaría bastante sospechoso. La explicación está en que el programa no coge asientos sino apuntes, es decir, las anotaciones sucesivas que se introducen desde el «comienzo de la sesión». Y por tanto no comprueba si el asiento está cuadrado o no (no sabe cuándo se acaba el asiento). Los datos de DEBE y HABER informan así del acumulado desde el inicio de las operaciones, información de dudosa utilidad. Con el dato sobre la diferencia se puede comprobar visualmente si ha habido algún error en la entrada de datos.

Apuntes	1000
Cuentas	3, 4, 5, 6 dígitos
Concepto	21 cts.
Importe	8 dígitos


DIARIO DE APUNTES AL : 07/10/84

PAGINA :2

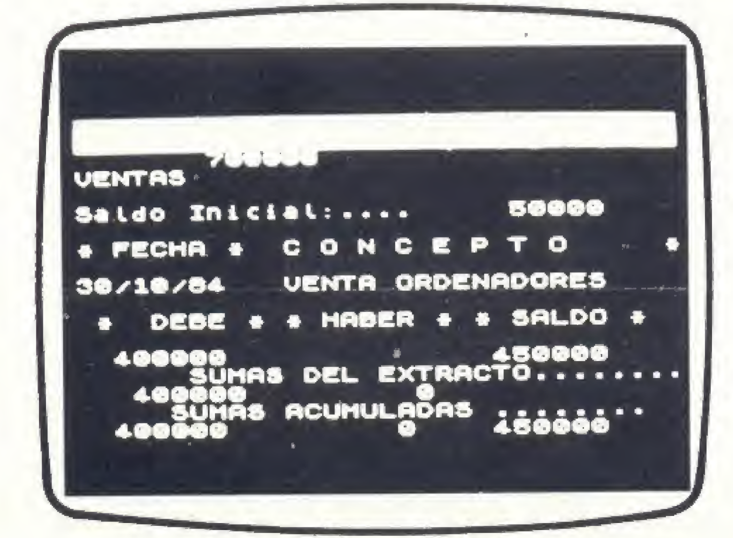
L I S T A D O   A P U N T E S

CUENTA	* TITULO *	* C O N C E P T O *	* M O V I M I E N T O S * DEBE *	* HABER *	Nro*	* FECHA *
47500 +	HACIENDA ACREE.CONC.FISC.	IRPF. 1-3/31-3 JULIO		4035	51	31/03/84
57000 +	CAJA	HABER.1-3/31-3 JULIO		33797	52	31/03/84
66100 +	COMUNICACIONES	TELEFONO 02 FEB/MARZO	42756		53	06/02/84
57200 +	BANCOS	PAGO TELEF.02 FEB/MAR		42756	54	06/02/84
66600 +	OTROS GASTOS	COF.S/S GYAS ENERO	2700		55	29/02/84
57200 +	BANCOS	PAGO S/S GYAS ENERO		2700	56	29/02/84
66600 +	OTROS GASTOS	CONF.S/S GYAS FEBRERO	2700		57	26/03/84
57200 +	BANCOS	PAGO S/S GYAS FEBRERO		2700	58	26/03/84
47700 +	ORGANISMOS S/S ACREEDORA	AUTON.S/S ENERO JUAN	11506		59	31/01/84
57200 +	BANCOS	PAGO AUTON.ENERO JUAN		11506	60	31/01/84
47700 +	ORGANISMOS S/S ACREEDORA	AUTON.S/S FEBRER.JUAN	11506		61	29/02/84
57200 +	BANCOS	PAGO AUTON.FEBRE.JUAN		11506	62	29/02/84
47700 +	ORGANISMOS S/S ACREEDORA	AUTON.S/S MARZO JUAN	11506		63	31/03/84
57200 +	BANCOS	PAGO AUTON.MARZO JUAN		11506	64	31/03/84
64200 +	SUMINISTROS	LUZ TARIFA 20 ENE/FEB	33151		65	10/01/84
57200 +	BANCOS	PAGO LUZ T.20 ENE/FEB		33151	66	10/01/84
64200 +	SUMINISTROS	LUZ TARIFA 30 ENE/FEB	88385		67	10/01/84
57200 +	BANCOS	PAGO LUZ T.30 ENE/FEB		88385	68	10/01/84
64200 +	SUMINISTROS	LUZ TARIFA 20 MAR/ABR	35127		69	06/03/84
57200 +	BANCOS	PAGO LUZ T.20 MAR/ABR		35127	70	06/03/84
64200 +	SUMINISTROS	LUZ TARIFA 30 MAR/ABR	88744		71	06/03/84
57200 +	BANCOS	PAGO LUZ T.30 MAR/ABR		88744	72	06/03/84
66600 +	OTROS GASTOS	CUOTA ACENOMA ENERO	1500		73	30/01/84
57200 +	BANCOS	PAGO ACENOMA ENERO		1500	74	30/01/84
66600 +	OTROS GASTOS	CUOTA ACENOMA FEBRERO	1500		75	29/02/84
57200 +	BANCOS	PAGO ACENOMA FEBRERO		1500	76	29/02/84
66600 +	OTROS GASTOS	CUOTA ACENOMA MARZO	1500		77	23/03/84
57200 +	BANCOS	PAGO ACENOMA MARZO		1500	78	23/03/84
64200 +	SUMINISTROS	PROPANO FACT.35315	86200		79	30/03/84
57200 +	BANCOS	PAGO PROPANO F.35315		86200	80	30/03/84
64100 +	REPARACION Y CONSERVACION	REPARA.J.MORA F.21318	12185		81	14/02/84
41000 +	ACREEDORES DIVERSOS	PAGO J.MORA FAC.21318		12185	82	14/02/84
41000 +	ACREEDORES DIVERSOS	PAGO J.MORA FAC.21318	12185		83	14/02/84
57000 +	CAJA	REPARA.J.MORA F.21318		12185	84	14/02/84
60000 +	COMPRAS	CONRAD.MARTIN F.75764	29469		85	30/03/84
40000 +	CONRADO MARTIN S.A.	PAGO FACTURA 75764		29469	86	30/03/84
40000 +	CONRADO MARTIN S.A.	PAGO FACTURA 75764	29469		87	30/03/84
57200 +	BANCOS	CONRAD.MARTIN F.75764		29469	88	30/03/84
60000 +	COMPRAS	CONRAD.MARTIN F.69945	33596		89	30/01/84
40000 +	CONRADO MARTIN S.A.	PAGO FACTURA 69945		33596	90	30/01/84
40000 +	CONRADO MARTIN S.A.	PAGO FACTURA 69945	33596		91	30/01/84
57200 +	BANCOS	CONRAD.MARTIN F.69945		33596	92	30/01/84
60000 +	COMPRAS	ANGEL DIAZ FACT.1733	1680		93	30/01/84
40001 +	ANGEL DIAZ (CERV.C.COLA)	PAGO FACTURA 1733		1680	94	30/01/84
40001 +	ANGEL DIAZ (CERV.C.COLA)	PAGO FACTURA 1733	1680		95	30/01/84
57000 +	CAJA	ANGEL DIAZ FACT.1733		1680	96	30/01/84
60000 +	COMPRAS	ANGEL DIAZ FACT.859	7500		97	02/02/84
40001 +	ANGEL DIAZ (CERV.C.COLA)	PAGO FACTURA 859		7500	98	02/02/84
40001 +	ANGEL DIAZ (CERV.C.COLA)	PAGO FACTURA 859	7500		99	02/02/84
57000 +	CAJA	ANGEL DIAZ FACT.859		7500	100	02/02/84
	S U M A Y S I G U E		1091354	1091354		

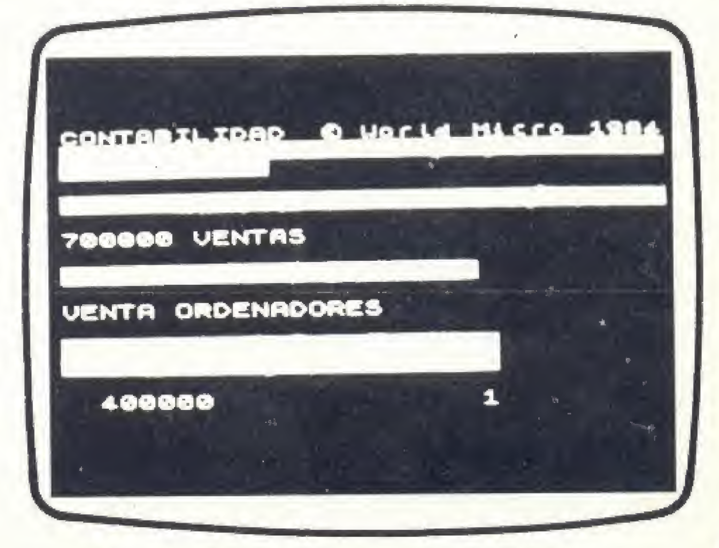
Listado de apuntes a una fecha. La opción por días, no incluye la última columna, y permite su ordenación por fechas.

Actualización

Este es uno de los capítulos más importantes a tener en cuenta en todo programa de contabilidad, consistente en actualizar el plan de cuentas con los datos de los asientos. Básicamente hay dos métodos: hacerlo directamente cada vez que se introduce un asiento o realizar este proceso en bloque como opción independiente. Esta segunda



Extracto de cuenta por pantalla.



Diario. Apuntes por pantalla.



opción suele ser la más empleada ya que no requiere la presencia del usuario, a excepción del momento inicial cuando se selecciona la opción de actualización. La actualización directa por asiento lleva un tiempo que resultaría excesivo para el *microdrive* y para la paciencia del usuario que había de estar «pegado» al ordenador todo el tiempo.

El programa utiliza esta segunda opción, tal y como hemos podido deducir del funcionamiento del programa, pues el manual de instrucciones sólo comenta la entrada de datos.

Al actualizar el plan de cuentas, en la pantalla van apareciendo los apuntes objeto de actualización. Tenga cuidado con la actualización. Normalmente salvo programas muy sofisticados sólo la podrá realizar una vez, es decir, que después de actualizar el plan de cuentas no podrá incorporar nuevos apuntes. Nosotros lo probamos, y nos quedamos en el intento: ¡el teclado se bloqueó!

"En precios hay uniformidad (aproximadamente 8.000 pts.). La prestaciones son radicalmente distintas."

Balances y extractos

Realizada la actualización, el siguiente paso es la obtención de balances y extractos. La opción tres nos da un nuevo submenú:

— Plan de cuentas.
— Balance de sumas y saldos.
— Estado del presupuesto.
— Diarios.
— Extractos de cuentas.

Sea cual fuese la opción elegida, todas ellas tienen dos características en común: se puede obtener los datos o varias cuentas o varios apuntes según los intervalos que se den y, en segundo lugar, se puede elegir la visualización por pantalla o por impresora. Esta segunda posibilidad resulta muy interesante para consultas rápidas y esporádicas.

Al estar realizado el programa en BASIC en su mayor parte es en la impresión donde se detectan los mayores tiempos de espera. No es una cuestión de mayor rapidez de impresión. Cada vez que hay que sacar el nombre de una cuenta se ha de recorrer la tabla de las 256 hasta que se localiza (mediante los bucles FOR-NEXT del BASIC).

Decíamos al inicio de este artículo que la mayor bondad de este programa era la capacidad de sus archivos. Pero también es cierto que ello conlleva el no poder dis-

UN CONTABLE MUY PARTICULAR

José María Oyarzábal lleva actualmente la contabilidad de treinta y cuatro clientes, gracias a sus tres ordenadores Spectrum a los que ha conectado distintas unidades de *microdrive*. Ahora confía en su Spectrum y en su programa de contabilidad, pero hace dos meses la situación no parecía tan clara: «Pedí presupuesto a varias casas. El precio de los equipos era de un millón a lo que había que sumar 100.000 pesetas del programa y el 10 por ciento de mantenimiento. Esto nos llevó a pensar en el Spectrum y a informarnos de los programas de contabilidad existentes para este ordenador. Vimos muchos y finalmente nos decidimos por el **World-Micro** por su capacidad. Hay otros más completos con opciones de regularización y reordenación por fechas, pero nos decidimos por éste porque permite trabajar con 256 cuentas y 1000 apuntes, requisito indispensable para el volumen de datos que manejamos».

Después de cuatro meses de trabajo intenso con el programa, José María sabe mucho de los problemas de implementación de todo programa nuevo. «Hasta hace una semana hemos estado subsanando errores». A mayor complejidad, mayores habrán de ser los períodos de prueba, que en este caso parece que fue llevada en gran parte por el cliente. «A los quinientos apuntes se bloqueaba. Tampoco se habían previsto listados de más de 50 cuentas y no se producía el salto de página... Eran pequeños problemas, pero ocasionaban una importante demora de tiempo, porque había que desplazarse a la tienda para que lo arreglasen».

Conscientes de los problemas que se dan en todo tipo de programa, y de que el factor tiempo jugaba en este caso un papel importante, aumentó su plantilla para incorporar una programadora que resolviese los problemas que se pudiesen dar. «Me sale rentable porque amplía los programas de acuerdo a nuestras necesidades específicas. Ahora hacemos nuestros propios programas».

«Lo que no habíamos previsto era la luz. ¡Nos volvemos locos!» Todavía con un brillo de asombro e incredulidad en sus ojos, José María nos contaba que un aumento en la tensión le fundió el interfa-

Potencie su ordenador al 100%

Nuevo Computer Monitor Philips V-7001.

A veces se realiza una buena inversión en la compra de un ordenador, y por comodidad, se adquiere un monitor de la misma marca sin pensar que, en muchos casos, el fabricante puede tener una gran experiencia en la fabricación de ordenadores pero escasa en la de monitores. El resultado de este tándem, ordenador de calidad y monitor de bajas prestaciones, puede convertir al ordenador en mucho menos capaz de lo que en realidad es, e injustificar el precio que se ha pagado por él.

Algo parecido ocurre con los usuarios de un ordenador doméstico que utilizan la pantalla del TV familiar. Con el agravante de que esta pantalla reduce la definición de los caracteres gráficos del ordenador, fatiga visualmente al poco tiempo de su utilización, distorsiona la imagen en sus contornos, sufre cierta incompatibilidad general y limita el uso del ordenador a las costumbres familiares de ver televisión.

Philips ahora lanza al mercado un nuevo producto que por sus características se adecúa eficazmente a esos casos, el Computer Monitor V-7001.

Un monitor que, al ser monochrome, se convierte en una pequeña inversión muy rentable.

Especialmente concebido para expresar datos alfanuméricos y gráficos generados por ordenador personal o doméstico.

Especialmente diseñado para aceptar una máxima compatibilidad con los ordenadores más difundidos.

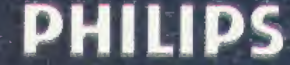
Con la garantía, calidad y fiabilidad del mayor fabricante de televisores y tubos de imagen del mundo.

Y con unas prestaciones muy por encima a las de otros monitores de su nivel:

- 2.000 caracteres, 80 caracteres por 25 líneas.
- Alta resolución en el tubo de imagen (18 MHz). Gran legibilidad.
- Alta tensión (12 KV). Imagen más brillante.
- Presentación de caracteres en fósforo verde P 31. Fósforo de reacción rápida que no deja rastro de manchas en la pantalla mientras se desarrolla.
- Pantalla tratada con antibrillo.
- Unidad de desviación especial de corrección magnética de la imagen.
- Doble posición ajustable.
- Salida de sonido.

Nuevo Computer Monitor V-7001. Pregunte a su distribuidor Philips o al establecimiento especializado más cercano.

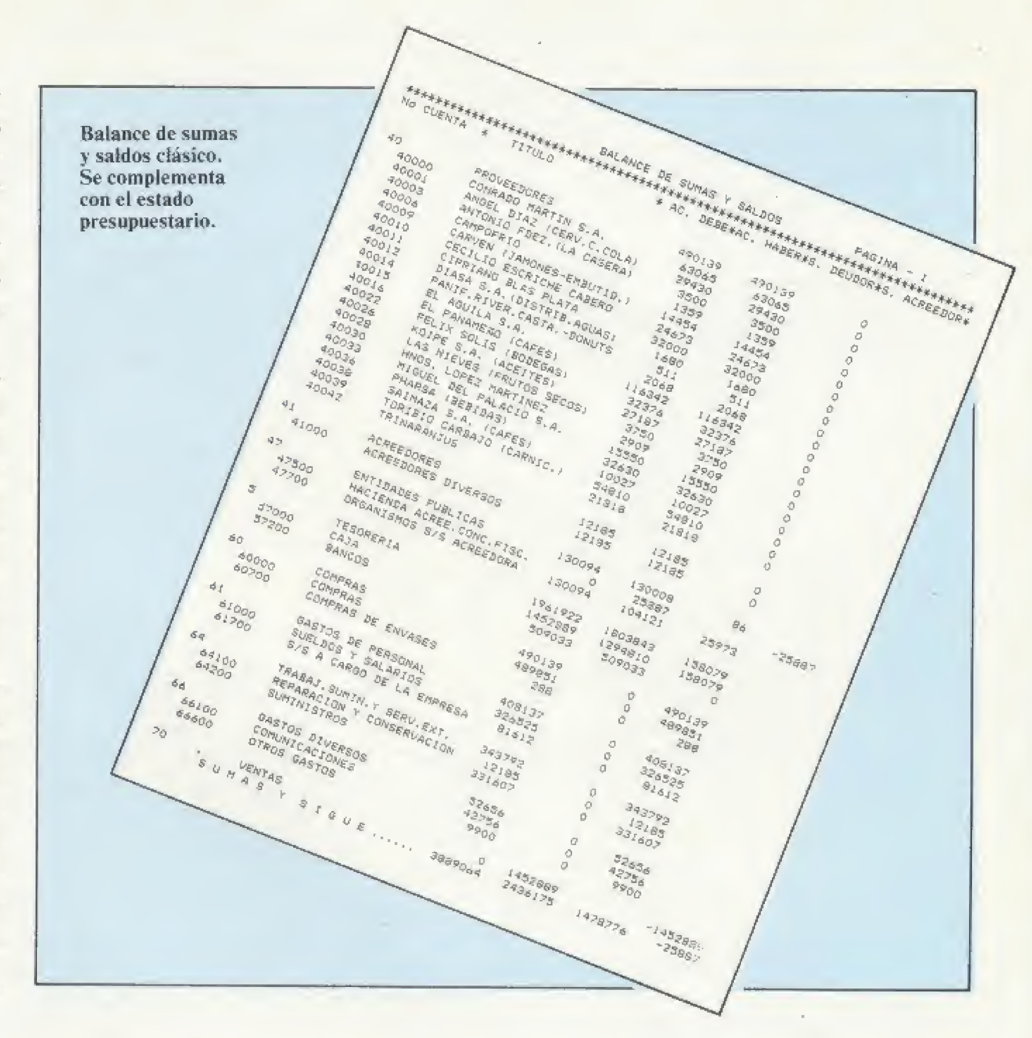






poner de la opción de regularización y el no contar con los apuntes ordenados por fechas. Esta segunda desventaja puede subsanarse obteniendo el diario «por fechas», pero ello enlentece la impresión y requiere la presencia del usuario para indicar las fechas en las que se desea obtenerlo.

Como todo programa, todavía es susceptible de mejoras. Pero también como todo programa, existen otros complementarios con distintas prestaciones que habrá de evaluar seriamente antes de hacer su elección. Y una última recomendación: no se fíe demasiado de pequeñas demostraciones con un número reducido de cuentas y apuntes. Cuando se llenan los archivos es fácil encontrar problemas que los programadores no pudieron ver por no trabajar a fondo con el programa y es igualmente fácil que los tiempos de acceso a la información e impresión se modifiquen sustancialmente y después parezca que sus programas no son tan rápidos como el de la tienda.



BALANCE DE SUMAS Y SALDOS — PAGINA - 1

No CUENTA	TITULO	AC. DEBE	AC. HABER	S. DEUDOR	S. ACREEDOR
40	PROVEEDORES	490139	490139	0	
40000	CONRADO MARTIN S.A.	63065	63065	0	
40001	ANGEL DIAZ (CERV.C.COLA)	29430	29430	0	
40003	ANTONIO FDEZ.(LA CASERA)	3500	3500	0	
40006	CAMPOFRIO	1359	1359	0	
40009	CARVEN (JAMONES-EMBUTID.)	14454	14454	0	
40010	CECILIO ESCRICHE CABERO	24673	24673	0	
40011	CIPRIANO BLAS PLATA	32000	32000	0	
40012	DIASA S.A.(DISTRIB.AGUAS)	1680	1680	0	
40014	PANIF.RIVER.CASTA.-DONUTS	511	511	0	
40015	EL AGUILA S.A.	2068	2048	0	
40016	EL PANAMEÑO (CAFES)	116342	116342	0	
40022	FELIX SOLIS (BODEGAS)	32376	32376	0	
40026	KOIPE S.A. (ACEITES)	27187	27187	0	
40028	LAS NIEVES (FRUTOS SECOS)	3750	3750	0	
40030	HNOS. LOPEZ MARTINEZ	2909	2909	0	
40033	MIGUEL DEL PALACIO S.A.	15550	15550	0	
40036	PHARSA (BEBIDAS)	32630	32630	0	
40038	SAIMAZA S.A. (CAFES)	10027	10027	0	
40039	TORIBIO CARBAJO (CARNIC.)	54810	54810	0	
40042	TRINARANJUS	21818	21818	0	
41	ACREEDORES	12185	12185	0	
41000	ACREEDORES DIVERSOS	12195	12185	0	
47	ENTIDADES PUBLICAS	130094	130008	86	
47500	HACIENDA ACREE.CONC.FISC.	0	25887		-25887
47700	ORGANISMOS S/S ACREEDORA	130094	104121	25973	
5	TESORERIA	1961922	1803843	158079	
57000	CAJA	1452889	1294810	158079	
57200	BANCOS	509033	509033	0	
60	COMPRAS	490139	0	490139	
60000	COMPRAS	489851	0	489851	
60700	COMPRAS DE ENVASES	288	0	288	
61	GASTOS DE PERSONAL	408137	0	408137	
61000	SUELDOS Y SALARIOS	326525	0	326525	
61700	S/S A CARGO DE LA EMPRESA	81612	0	81612	
64	TRABAJ.SUMIN.Y SERV.EXT.	343792	0	343792	
64100	REPARACION Y CONSERVACION	12185	0	12185	
64200	SUMINISTROS	331607	0	331607	
66	GASTOS DIVERSOS	52656	0	52656	
66100	COMUNICACIONES	42756	0	42756	
66600	OTROS GASTOS	9900	0	9900	
70	VENTAS	0	1452889		-1452889
	S U M A S Y S I G U E	3889044	2436175	1478776	-25887

Balance de sumas y saldos clásico. Se complementa con el estado presupuestario.



ce 1, la fuente de alimentación y el Spectrum. En otra ocasión la inestabilidad del fluido eléctrico había supuesto la pérdida de un día de trabajo, aproximadamente 8000 apuntes perdidos. «Compramos una fuente de alimentación especial para estabilizar la corriente y no hemos vuelto a tener problemas».

«No nos preocupa que no regularice o que no haga el cierre de cuentas. Esto lo hacemos nosotros aparte. La mayor limitación que le encontramos es que no reordena por fechas, al menos directamente, lo que subsanamos tirando listados de diarios por fechas, pero esto es lento, se necesitan dos días por cliente. Tenemos otros programas que realizan esta ordenación, pero el problema es siempre el mismo: el número de apuntes se reduce en un 75 por ciento. A nosotros esto no nos interesa.»



AGENDA:

La otra forma de anotar sus compromisos

Comencemos por plantearnos un problema modesto. Queremos obtener el día de la semana en que cae el 1 de enero de cualquier año teniendo para ello en cuenta que en 1984 el 1 de enero fue domingo. Como se podrá suponer, si logramos resolver esto habremos dado un gran paso en nuestro intento de obtener el calendario del año requerido. Ante la magnitud de la empresa, queremos ahora encontrar en qué cae el 1 de enero de 1985. Como 1984 es bisiesto, tiene 366 días (febrero tiene 29), necesitamos encontrar el número de semanas que van de 1 a 1 de enero. Para ello calculamos.

INT (366/7) = 52

Como 52 × 7 = 364 nos damos cuenta que nos faltan dos días para completar el año y que en consecuencia el 1 de enero de 1985 cae en martes. Dudo mucho que se haya perdido en la argumentación anterior, pero si no es así léala otra vez, pues éste es el truco esencial del programa.

Volvamos ahora con nuevos ánimos a calcular el día 1 de cualquier año tomando como punto de partida el 1 de enero de 1984. Para hacer esto, debemos aún vencer la pequeña dificultad que significa que no todos los años tengan el mismo número de días. Para superarla actuemos de la siguiente forma: cada 4 años a partir del 84 nos vamos a encontrar con otro bisiesto, es decir, tomaremos como período natural de tiempo un intervalo de 1461 días (4 × 365 + 1) con

lo que dado un año cualquiera para hallar el número de días que van del 1 de enero de 1984 al 1 de enero de este año que llamaremos X calculamos:

INT((X–1984)/4)×1461 (1)

Que nos da el número de días que hay entre dos años bisiestos sucesivos, y a continuación calculamos:

(X – 1984 – INT((X–1984)/4)×4)× ×365 (2)

Que nos da el número de días que hay desde el último año bisiesto hasta el considerado. Sumando esto a (1) obtendremos el resultado apetecido.

Supóngase que queremos hallar el número de días que hay del 1 de enero de 1984 al 1 de enero de 2002. Comenzamos por calcular (1).

INT ((2002–1984)/4)= 4; 4×1461= 5844

En consecuencia hemos encontrado que en el período considerado hay 4 grupos de 4 años y en estos 4 grupos un total de 5844 días.

Calculamos ahora (2).

2002 – 1984 – 4 × 4 = 2; 2 × 365 = 730

Por tanto el número de días será 5844 + 730 = 6574

Calculemos ahora el número de semanas del siguiente modo:

INT (6574/7)= 939

Y ahora calculemos

6574 – 939×7= 1

En consecuencia, de 1 de enero de 1984 a 1 de enero de 2002 sobra un día para que el número de se-

```
  10 DIM L(12,31): DIM C(12,31)
 : DIM a(12,31): DIM m$(12,13): D
 IM d$(7,12)
  20 LET Z$="": LET DIAM=0: LET
 C$="I/7-INT (I/7)": LET A$="DIF/
 4-INT (DIF/4)": LET B$="INT (DI/
 7)"
 100 LET m$(1)="ENERO       31":
 LET M$(2)="FEBRERO    28": LET M
 $(3)="MARZO      31"
 110 LET M$(4)="ABRIL       30":
 LET M$(5)="MAYO        31": LET M
 $(6)="JUNIO       30"
 120 LET M$(7)="JULIO       31":
 LET M$(8)="AGOSTO      31": LET M
 $(9)="SEPTIEMBRE 30"
 130 LET M$(10)="OCTUBRE     31":
 LET M$(11)="NOVIEMBRE  30": LET
 M$(12)="DICIEMBRE  30"
 140 LET D$(1)="LUNES": LET D$(2
 )="MARTES": LET D$(3)="MIERCOLES
 ": LET D$(4)="JUEVES": LET D$(5)
 ="VIERNES": LET D$(6)="SABADO":
 LET D$(7)="DOMINGO"
 150 LET a(1,1)=1: LET a(1,6)=1:
 LET a(3,19)=1: LET a(4,1)=1: LE
 T a(6,21)=1: LET a(7,25)=1: LET
 a(8,15)=1: LET a(10,12)=1: LET a
 (11,1)=1: LET a(12,25)=1
 500 CLS : PRINT "QUIERE:": PRIN
 T : PRINT "0 - OBTENER UN ANO":
 PRINT "1 - OBTENER UN MES"
 510 PRINT "2 - HACER ANOTACIONE
 S"
 520 PRINT "3 - VER ANOTACIONES"
 530 PRINT "4 - CANCELAR ANOTACI
 ONES"
 535 PRINT "5 - CORREGIR ANOTACI
 ONES"
 540 PRINT "6 - GRABAR ANOTACION
 ES"
 550 PRINT "7 - CARGAR ANOTACION
 ES"
 560 INPUT NU
 565 IF NU=0 THEN  GO TO 1000
 570 IF NU=1 THEN  GO TO 1000
 580 IF NU=2 THEN  GO TO 2000
 590 IF NU=3 THEN  GO TO 2500
 600 IF NU=4 THEN  GO TO 3000
 610 IF NU=5 THEN  GO TO 3500
 620 IF NU=6 THEN  GO TO 4000
 630 IF NU=7 THEN  GO TO 4500
1000 INPUT "ANO: ";ANO
1005 IF nu=0 THEN  GO TO 8000
1020 IF nu<>0 THEN  CLS : GO SUB
 9000: INPUT "MES: ";MES
1030 LET DIF=ANO-1984
1040 LET DIA=INT (DIF/4)*1461
1050 LET RES=(DIF-INT (DIF/4)*4)
*365: LET DIA1=DIA+RES
1060 LET DIAM=0: FOR I=1 TO MES-
1
1065 LET DIAM=DIAM+VAL M$(I,12 T
O 13)
1067 IF VAL A$=0 AND I=2 THEN  L
ET DIAM=DIAM+1
1068 NEXT I
1070 LET DI=DIA1+DIAM: LET NOM=I
NT (DI/7): LET RED=DI-7*NOM
1075 IF VAL A$<>0 THEN  LET RED=
RED+1
1080 IF RED=0 THEN  LET RED=7
1090 LET RED=ABS RED: PRINT D$(R
ED)
1095 IF nu=0 THEN  RETURN
1100 GO TO 8000
2000 CLS : GO SUB 9000
2005 INPUT "MES (N PARA SALIR):
";P$: IF P$="N" OR p$="n" THEN
GO TO 500
2007 LET M=VAL P$
2010 INPUT "DIA: ";D
```

manas sea exacto y por tanto será lunes (recuérdese que el 1 de enero del 84 fue domingo).

Esto está bien, pero no es nuestro propósito, queremos conseguir, dado un mes, el calendario de éste. Esto significa saber en qué día de la semana empieza el mes, lo cual a su vez significa obtener el número de días que van del 1 de enero al 1 del mes considerado. Para resolver este problema optamos por la siguiente solución: en una matriz m$(12,31) introducimos el nombre del mes y en el carácter 12 y 13 del elemento de la matriz, introducimos el número de días del mes (véase líneas 110 a 130). Ahora ya es fácil resolver nuestro problema. Imprimimos en la pantalla los meses con sus respectivos números (a enero le corresponde 1, a febrero 2...) y pedimos el año y mes del que se quiere el calendario.

(1) LET DIAM= 0: FOR I=1 TO MES–1
(2) LET DIAM= DIAM + M$(I,12 TO 13)
(3) IF VAL A$=0 AND I=2 THEN LET DIAM= DIAM + 1
(4) NEXT I

A$ está definida por A$= "DIF/4 – INT (DIF/4)" siendo DIF la diferencia entre 1984 y el año deseado. A la luz de estas definiciones debe estar claro el propósito de la línea (3) (1067 en el programa), para detectar si el año es bisiesto o no (como empezamos en el 1984, que sí lo es, cada vez que pasen 4 años también lo será y VAL A$ será 0 en estos años) y en el caso de que lo sea, al mes de fe-

```
2015 CLS : PRINT  FLASH 1;"PREPA
RADO PARA ANOTACACION"
2020 INPUT  LINE F$
2030 LET LONG=LEN F$: LET COM=LE
N Z$: LET L(M,D)=LONG: LET C(M,D
)=COM
2040 LET Z$=Z$+F$
2100 BEEP 1,7: CLS : PRINT  BRIG
HT 1; FLASH 1;"ANOTADO (PULSE UN
A TECLA)": PAUSE 0: GO TO 2000:
GO TO 500
2500 CLS : GO SUB 9000
2504 INPUT "MES(N PARA SALIR): "
;P$: IF P$="N" OR P$="n" THEN  G
O TO 500
2505 LET M=VAL P$
2510 INPUT "DIA: ";D
2515 IF L(M,D)=0 THEN  CLS : BEE
P 1,2: PRINT  FLASH 1;"EN ESTA F
ECHA NO HAY ANOTACION. REPITA DA
TOS POR FAVOR": PAUSE 150: GO TO
 2500
2520 LET LONG=L(M,D)-1: LET COM=
C(M,D)+1
2530 CLS : PRINT Z$(COM TO COM+L
ONG)
2535 IF NU=5 OR NU=4 THEN  RETUR
N
2540 PRINT  BRIGHT 1;#1;"TERMINA
DO PULSE UNA TECLA": PAUSE 0: GO
 TO 2500
3000 GO SUB 2500
3010 LET C(M,D)=0: LET Z$=Z$(1 T
O COM-1)+Z$(COM+LONG+1 TO LEN Z$
)
3015 PRINT AT 21,0;"ESPERE UN MO
MENTO."
3020 FOR W=M TO 12: FOR J=1 TO 3
1
3030 IF C(W,J)<>0 THEN  LET C(W,
J)=C(W,J)-(LONG+1)
3040 NEXT J: NEXT W
3050 LET L(M,D)=0: BEEP .5,7: PR
INT #1;"VALE": PAUSE 50: GO TO 3
000
3500 LET BO=0: GO SUB 2500
3510 INPUT "N PARA SALIR: ";X$:
IF X$="N" OR X$="n" THEN  GO TO
500
3515 PRINT : FOR I=COM TO COM+LO
NG
3520 PRINT  INVERSE 1;Z$(I);: BE
EP .1,7
3530 IF INKEY$=" " THEN  INPUT
LINE R$: LET Z$=Z$(1 TO I)+R$+Z$
(I+1 TO LEN z$): GO SUB 9700: GO
 SUB 2530: GO TO 3510
3533 IF I=LONG+COM AND BO=1 THEN
  LET BO=0: PRINT "#";: INPUT "S
EGURO? S/N: ";W$: GO SUB 9990: I
F W$="S" OR W$="s" THEN  GO SUB
9500: GO TO 3510
3535 IF CODE INKEY$=13 AND BO=0
THEN  LET BO=1: PRINT "#";: LET
IS=I: BEEP 1,7
3537 IF CODE INKEY$=13 AND BO=1
THEN  LET BO=0: PRINT "#";: INPU
T "SEGURO? S/N: ";W$: GO SUB 999
0: IF W$="S" OR W$="s" THEN  GO
SUB 9500: GO TO 3510
3538 IF INKEY$<>"M" THEN  GO TO
3530
3550 NEXT I
3560 GO TO 3510
4000 INPUT "NOMBRE DEL FICHERO:
";X$
4010 SAVE X$ DATA Z$(): SAVE X$
DATA L(): SAVE X$ DATA C()
4020 GO TO 500
4500 INPUT "NOMBRE DEL FICHERO:
";X$
4510 IF CODE X$=13 THEN  LOAD ""
```

brero (I=2) le sumamos 1 día de modo que tenga 29 y no 28 días.

Cuando se han ejecutado estas líneas sumamos el número de días obtenido al número de días que van de 1 de enero a 1 de enero, calculamos el número de semanas que hay en ese intervalo y el resto de la división: n.º de días totales/7. Con esto sabemos en qué día de la semana comienza el mes, acabando la tediosa parte de cálculo.

Lo siguiente es imprimir el mes deseado. Para ello comenzamos imprimiendo en inverso el mes y año de que se trata, y a continuación los días de la semana. Después definimos la variable FIN= VAL A$(MES,12 TO 13) que nos indica la longitud del mes con la salvedad ya mencionada de que si el año es bisiesto y el mes febrero FIN= FIN + 1. A continuación con un bucle FOR I=1 TO FIN vamos colocando los días uno a uno con la ayuda del comando AT CON-,CO donde CON y CO son dos variables que indican el número de línea (incrementándose en dos cada vez que CO=25) y el número de columna (incrementándose de 4 en 4 hasta llegar a 25 momento en el cual se inicializa a 1), respectivamente.

"El mes, con su número de días, se mete en una tabla."

Para terminar con esta sección de programa tan sólo aclarar tres puntos:

1) La variable CO debe de estar inicializada para que el primer día del mes caiga en el día anteriormente hallado. De esta forma, si por ejemplo el día uno es martes, el resto de dividir el número de días entre 7 va a ser 2 y en conse-

```
 DATA S$(): LOAD "" DATA L(): LO
AD "" DATA C(): GO TO 500
4520 LOAD X$ DATA S$(): LOAD X$
DATA L(): LOAD X$ DATA C(): GO T
O 500
8000 IF nu=0 THEN  FOR j=1 TO 12
: LET mes=j: GO SUB 1030
8005 CLS : PRINT  INVERSE 1;M$(M
ES,1 TO 11),ANO: PRINT : PRINT "
 L   M   X   J   V   S   D": PRI
NT
8010 LET FIN=VAL M$(MES,12 TO 13
)
8015 IF MES=2 AND VAL A$=0 THEN
 LET FIN=FIN+1
8020 LET CO=(RED-1)*4+1: LET CON
=4: FOR I=1 TO FIN
8023 IF L(MES,I)<>0 AND ANO=1985
 AND i<=9 THEN  PRINT AT CON,CO-
1;"[";AT CON,CO+1;"]"
8024 IF L(MES,I)<>0 AND ANO=1985
 AND i>=9 THEN  PRINT AT CON,CO-
1;"[";AT CON,CO+2;"]"
8025 IF a(mes,i)=1 AND co<>25 TH
EN  PRINT  INVERSE 1;AT con,co;i
: LET co=co+4: GO TO 8050
8030 IF co<>25 THEN  PRINT AT CO
N,CO;I: LET CO=CO+4: GO TO 8050
8040 IF CO=25 THEN  PRINT  INVER
SE 1;AT CON,25;i: LET CON=CON+2:
 LET CO=1
8050 NEXT I
8055 IF NU=0 THEN  PRINT #1;"PUL
SE UNA TECLA (R PARA RETROCE_DER
)": PAUSE 0: IF INKEY$="R" OR IN
KEY$="r" THEN  LET j=j-2
8057 IF nu=0 THEN  NEXT j
8060 PRINT #1;"TERMINADO (PULSE
UNA TECLA)": PAUSE 0: GO TO 500
9000 FOR I=1 TO 12: PRINT TAB 0;
I;TAB 4;" - ";TAB 8;M$(I): NEXT
I
9010 RETURN
9500 LET ERU=LEN Z$: LET Z$=Z$(1
 TO IS)+Z$(I+1 TO ERU)
9510 LET LONG=LONG-(I-IS): LET L
(M,D)=LONG+1
9515 CLS : PRINT Z$(COM TO LONG+
COM)
9517 PRINT AT 21,0;"ESPERE UN MO
MENTO."
9520 FOR W=1 TO 12: FOR J=1 TO 3
1
9530 IF C(W,J)<>0 AND C(W,J)>C(M
,D) THEN  LET C(W,J)=C(W,J)-(I-I
S)
9540 NEXT J: NEXT W
9550 RETURN
9700 LET URU=LEN R$: LET L(M,D)=
L(M,D)+URU: LET LONG=LONG+URU
9705 PRINT AT 21,0;"ESPERE UN MO
MENTO."
9710 FOR W=1 TO 12: FOR J=1 TO 3
1
9720 IF C(W,J)<>0 AND C(W,J)>C(M
,D) THEN  LET C(W,J)=C(W,J)+URU
9730 NEXT J: NEXT W
9740 PRINT Z$(COM TO COM+LONG)
9750 RETURN
9990 IF W$="N" OR W$="n" THEN  C
LS : PRINT Z$(COM TO COM+LONG):
PRINT : PRINT  INVERSE 1;Z$(COM
TO IS);: LET i=is
9995 RETURN
9997 IF CODE INKEY$=0 THEN  GO T
O 9997
9999 PRINT CODE INKEY$: GO TO 99
90
```

cuencia deberemos comenzar a imprimir en (2 – 1)×4 + 1 (el lunes que tiene por resto 1 lo colocamos en la columna 1) como se puede ver en la línea 8020.

2) Cada vez que CO=25 estaremos en domingo y como es fiesta queremos que se imprima en inverso (línea 8040).

3) Para detectar una fiesta dimensionamos una matriz a(12,31) en la cual ponemos a 1 los elementos que son fiesta.

Obtener el calendario de un año entero se limita a añadir la línea 8000 y 8057 que lo único que hacen es ir recorriendo el calendario de mes en mes.

La segunda parte del programa (opciones 2, 3, 4, 5, 6) convierten al Spectrum en una verdadera agenda. Con estas opciones, podrá hacer anotaciones, verlas, corregirlas, tacharlas, grabarlas y cargarlas de cinta. En otros lenguajes con ficheros, esta parte del programa estaría prácticamente resuelta, pero no es así en BASIC que no se dispone de ficheros propiamente dichos. Un primer modo de atacar el problema consistiría en definir una matriz de cadena en la cual cada elemento viniera definido por el mes y el día del mes en que se quiere hacer la anotación. Proceder de este modo es inadecuado, pues entonces estamos obligados a dar una longitud máxima a la anotación. Como las anotaciones pueden tener tamaños muy distintos limitaríamos muy seriamente la longitud máxima o desperdiciaríamos una gran cantidad de memoria. La solución dada al problema se puede dividir en dos puntos.

1) Meter todas las anotaciones en una variable de cadena Z$ en la que no hay que especificar su longitud.

2) Para poder localizar y leer cada una de las anotaciones, dimensionar dos matrices L(12,31) y C(12,31) en la que guardar las longitudes y comienzo de la anotación correspondiente al día (i,j) en la matriz Z$.

Este método tiene la ventaja de ahorrar una gran cantidad de memoria, pero también tiene el inconveniente de que los procesos de localización y lectura se hacen más

"Para las anotaciones se utiliza una variable de cadena."

complicados. Veamos cómo funciona en un caso práctico. Supongamos que el día 1 de junio tenemos un examen de física con lo que la anotación es: «Examen de física».

Supongamos ahora que el 10 de marzo es el cumpleaños de una amiga, que llamaremos María, la anotación será: «Cumpleaños de María. Comprar regalo».

En Z$ tendremos.

Z$= «Examen de física. Cumpleaños de María. Comprar regalos».

Y las matrices auxiliares tendrán todos sus elementos a cero salvo la de los días en que se ha anotado algo:

C(6,1)= 1 // L(6,1)=17
C(3,10)=18 // L(3,18)=37

Cuando queramos recuperar el mensaje de un día cualquiera meteremos en el ordenador la fecha de este día e inmediatamente obtendremos el mensaje, en caso de haberlo. Para esto el ordenador define las variables: COM= C(i,j), LONG= L(i,j), e imprime Z$(COM TO COM+LONG) en pantalla.

Este es a grandes rasgos el proceso de creación de notas, desde luego falta comentar algunos detalles técnicos de menor importancia como es el hecho de que la introducción del mensaje se hace a través de una variable de cadena Z$ mediante la cual se define L(i,j) como: L(i,j)= LEN F$, pudiéndose definir entonces C(i,j) según: C(i,j)= LEN Z$, y definir por último Z$ como: Z$= Z$ + F$

La siguiente opción en el menú es la referida a la eliminación de anotaciones. Se extiende desde la línea 3000 a la 3050 y comienza haciendo.

Z$ = Z$(1 TO COM – 1) + Z$(COM+LONG+1 TO LEN Z$)

Donde tanto COM como LONG están definidas para la anotación a eliminar. Una vez conseguida la desaparición de la anotación deseada de Z$, lo que conviene hacer ahora es poner tanto el elemento de matriz C(i,j) como el L(i,j) a cero con lo que el ordenador interpretará en lo sucesivo que en el día definido por el mes "i" y el día de mes "j" no hay ninguna anotación. Ocurre ahora que al desaparecer este elemento ha quedado un hueco dentro de la matriz C(i,j) que define el comienzo de cada mensaje. Para cerrarlo utilizamos el bucle de las líneas 3020 a 3040 que lo que hace es restar a todos los elementos de matriz cuyo comienzo es superior al que hemos quitado la longitud del mensaje eliminado.

Veamos esto con un ejemplo. Supongamos que aparte de los dos mensajes anteriores, tenemos un tercero para el día 17 de agosto que dice así: «Regar las plantas». Con lo que ahora Z$ queda:

Z$= «Examen de física. Cumpleaños de María. Comprar regalos. Regar las plantas».

C(8,17)= 74 // L(8,17)= 18

Si ahora suprimimos la segunda anotación deberemos, siguiendo el texto anterior, poner:

Z$= Z$(1 TO COM – 1) + Z$(COM+LONG+1 TO LEN Z$)

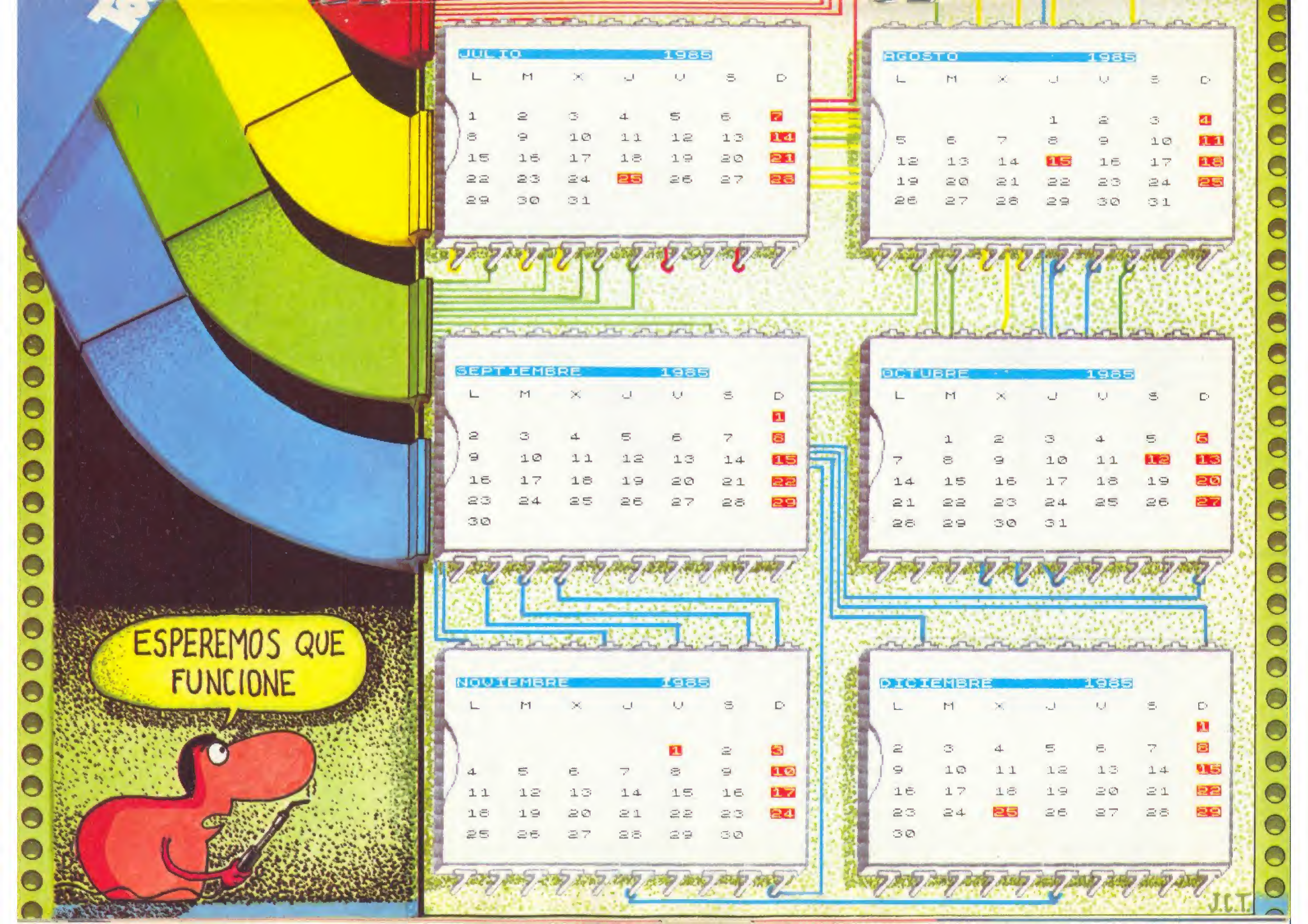
Z$ = Z$(1 TO 18) + Z$(56 TO 73)

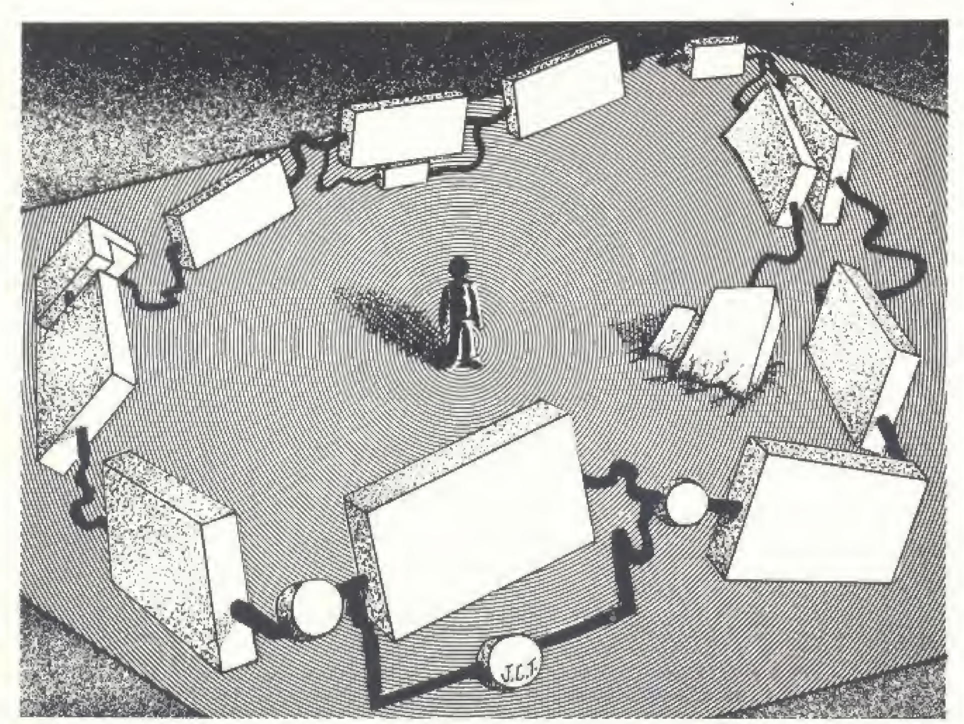
Entonces el bucle anteriormente citado sólo actuará sobre C(8,17) que quedará ahora C(17,18)= 74 – 28= 46.

Mediante la opción 5, es posible corregir cualquier anotación. Para ello, después de haber introducido la fecha requerida se obtendrá el texto a corregir. Apretando sucesivamente la «m» se obtiene letra a letra el texto en video Inverso. Cuando llegue al comienzo de una palabra o grupo de palabras que quiera suprimir pulse ENTER con lo que obtendrá el carácter «#», a continuación pulse de nuevo la «m» hasta llegar al final del texto a suprimir. Pulse de nuevo ENTER, obteniendo otro «#». Una vez hecho esto, puede ejecutarlo tecleando una «s» o cancelarlo pulsando una «n». Si lo que quiere es añadir un grupo de palabras pulse «m» hasta llegar al lugar donde quiera insertar las nuevas palabras. Pulse SPACE y escriba seguidamente el nuevo texto.

Julio Gómez Herrero


1985
ENERO 1985
L M X J V S D
1 2 3 4 5 6
7 8 9 10 11 12 13
14 15 16 17 18 19 20
21 22 23 24 25 26 27
28 29 30 31
FEBRERO 1985
L M X J V S D
1 2 3
4 5 6 7 8 9 10
11 12 13 14 15 16 17
18 19 20 21 22 23 24
25 26 27 28
MARZO 1985
L M X J V S D
1 2 3
4 5 6 7 8 9 10
11 12 13 14 15 16 17
18 19 20 21 22 23 24
25 26 27 28 29 30 31
ABRIL 1985
L M X J V S D
1 2 3 4 5 6 7
8 9 10 11 12 13 14
15 16 17 18 19 20 21
22 23 24 25 26 27 28
29 30
MAYO 1985
L M X J V S D
1 2 3 4 5
6 7 8 9 10 11 12
13 14 15 16 17 18 19
20 21 22 23 24 25 26
27 28 29 30 31
JUNIO 1985
L M X J V S D
1 2
3 4 5 6 7 8 9
10 11 12 13 14 15 16
17 18 19 20 21 22 23
24 25 26 27 28 29 30


JULIO 1985
L M X J V S D
1 2 3 4 5 6 7
8 9 10 11 12 13 14
15 16 17 18 19 20 21
22 23 24 25 26 27 28
29 30 31
AGOSTO 1985
L M X J V S D
1 2 3 4
5 6 7 8 9 10 11
12 13 14 15 16 17 18
19 20 21 22 23 24 25
26 27 28 29 30 31
SEPTIEMBRE 1985
L M X J V S D
1
2 3 4 5 6 7 8
9 10 11 12 13 14 15
16 17 18 19 20 21 22
23 24 25 26 27 28 29
30
OCTUBRE 1985
L M X J V S D
1 2 3 4 5 6
7 8 9 10 11 12 13
14 15 16 17 18 19 20
21 22 23 24 25 26 27
28 29 30 31
NOVIEMBRE 1985
L M X J V S D
1 2 3
4 5 6 7 8 9 10
11 12 13 14 15 16 17
18 19 20 21 22 23 24
25 26 27 28 29 30
DICIEMBRE 1985
L M X J V S D
1
2 3 4 5 6 7 8
9 10 11 12 13 14 15
16 17 18 19 20 21 22
23 24 25 26 27 28 29
30
ESPEREMOS QUE FUNCIONE
J.C.T.



Descubrimiento de un nuevo lenguaje: PASCAL

Hasta ahora siempre hemos utilizado nuestro Spectrum para programar con él en BASIC, o en código máquina, pero existen otras posibilidades de trabajar con otros lenguajes de programación, desconocidos para la mayoría. En este artículo vamos a tratar de introducirnos un poco en el lenguaje PASCAL.

Para ello trataremos de explicar las características esenciales de este lenguaje, las instrucciones más importantes y los diagramas de construcción, intentando ilustrarlo con pequeños programas ejemplo.

No pretendemos que esto sea un curso intensivo de PASCAL, sino dar una introducción para todo aquel que se sienta interesado en sacar un mayor rendimiento a su Spectrum, y a programar en otros lenguajes distintos de los habituales.

Lo primero que debemos indicar al respecto, es que para poder utilizar el lenguaje PASCAL, nos será necesario un compilador. Existen varios modelos en el mercado, cada uno con sus características especiales. Una vez en nuestro poder el compilador, estaremos en condiciones de poder adentrarnos en el mundo del PASCAL.

Características del PASCAL

La primera versión del lenguaje de programación PASCAL nace en el año 1968, siguiendo la línea de los lenguajes Algol 68, y Algol W. El primer compilador operativo aparece en el año 1970, siendo publicado en 1971.

El gran interés que provocaron estas publicaciones, motiva el desarrollo de otros compiladores, y como consecuencia la consolidación de este lenguaje. Por último, como iniciativa de la BCS circularon entre los años 1978 y 1980, cinco borradores del standard de PASCAL, consiguiéndose así su standarización. Así el PASCAL nace como consecuencia del hecho de ir buscando lenguajes de programación cada vez más adecuados al usuario, dándole un lenguaje estructurado y de una complejidad baja.

Las características del lenguaje PASCAL coinciden con las de la Programación Estructurada que son principalmente las siguientes:

a) Que el programa sea secuencial.

b) Que sea estructurado, es decir, que se pueda escribir el programa sólo con las siguientes estructuras básicas: proceso secuencial, IF-THEN-ELSE, y el bucle WHILE (que serán explicadas más adelante).

c) Que sea corto.

d) Que esté fragmentado. Se intenta conseguir que el programa esté dividido en trozos, para su mejor mantenimiento y comprensión.

Una vez hecha esta breve reseña histórica, pasamos directamente a explicar las características de este lenguaje.

1. Juego de caracteres

El juego de caracteres de que se compone este lenguaje es el siguiente:

— Caracteres numéricos: Comprenden los dígitos del 0 al 9.

— Caracteres alfabéticos: Están incluidas las letras de la A a la Z.

— Caracteres especiales: En ellos están incluidos los signos de puntuación y todos los tipos de operadores.

La combinación de todos los caracteres indicados mediante ciertas reglas de formación constituye los elementos integrantes de un programa PASCAL.

2. Elementos de un programa

El Pascal como todos los lenguajes tiene sus propias palabras reservadas y distintos identificadores.

Las palabras reservadas son aquellas que, como todos sabemos, pertenecen al vocabulario del propio lenguaje.

Los identificadores pueden ser de dos tipos:

a) Standard: Ya declarados en el propio lenguaje, siendo las propias palabras reservadas.

b) No standard: Los crea el programador. Para formarlos se usan caracteres alfabéticos y numéricos, con la única regla de que

el primer carácter deberá ser una letra. Por ejemplo: A14V sería correcto, 1ABC no sería correcto por ser el primer carácter un dígito.

Los operadores de que disponemos son de tres tipos:

a) Aritméticos: indican operaciones aritméticas. Son:

+, –, *, /, DIV, MOD

Donde: / indica división entre números reales; DIV indica división entre números enteros, y MOD calcula el resto de la división.

b) De relación: Realizan operaciones de comparación entre datos. Se utilizan igual que en BASIC. Son:

>, <, =, ⩾, ⩽, < >,

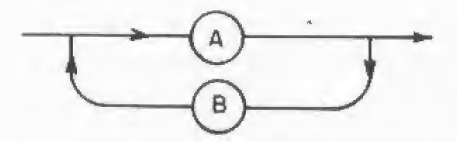
c) Lógicos: Se utilizan de la misma forma que en BASIC, y son AND, OR, NOT.

d) Para utilización de conjuntos: Se verá en su momento.

3. Estructura gramatical de un programa

La estructura gramatical del programa y de cada una de las instrucciones que lo componen, viene dada por lo que se llaman los Diagramas de Conway. Veamos un ejemplo de cómo seguir estos diagramas:

Mensaje 1:



La tira de caracteres ABA es un mensaje correcto. En cambio la tira ABB no es un mensaje autorizado, ya que no se puede seguir mediante las líneas del diagrama.

Una vez visto el funcionamiento de los diagramas veamos su especificación. Existen tres clases de símbolos:

Indica palabra reservada

Indica un operador

Hace referencia a otros diagramas sintácticos

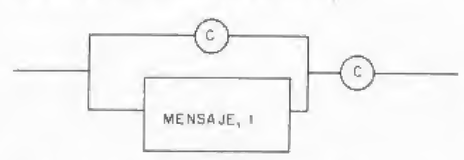
```
AB47    10 PROGRAM EJEMPLO;
AB47    20 CONST
AB47    30    N=3;
AB47    40 VAR
AB50    50    A,B,C,SUMA:INTEGER;
AB50    60    MEDIA:REAL;
AB50    70 BEGIN
AB59    80    READ(A,B,C);
AB68    90    SUMA:=A+B+C;
AB87   100    MEDIA:=SUMA/N;
AB9F   105 WRITE(CHR(16));
ABA6   110    WRITELN('LA SUMA DE
',A,' ,',B,'Y ',C,'ES ',SUMA);
AC03   120    WRITELN;
AC06   130    WRITELN('LA MEDIA AR
ITMETICA ES ',MEDIA:7:3);
AC3F   140    WRITELN;
AC42   150 END.
End Address: AC44

Run?

LA SUMA DE 345 ,543 Y 32 ES 920

LA MEDIA ARITMETICA ES 306.667
```

Un ejemplo de esta referencia a otros diagramas podría ser:



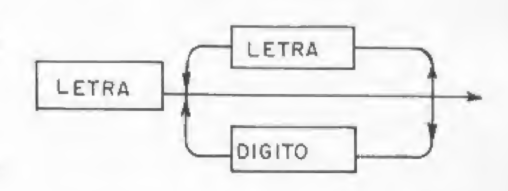
La tira ABAC se produce sustituyendo en el cuadrado el mensaje 1 que explicamos anteriormente. La estructura de un programa PASCAL en el caso más genérico es la siguiente:

PROGRAM nombre
LABEL lista de etiquetas
CONST declaración de constantes.
TYPE definición de tipos.
VAR definición de variables.
PROCEDURE nombre del procedimiento.
FUNCTION nombre de función.
BEGIN.
Estructura del programa.
END.

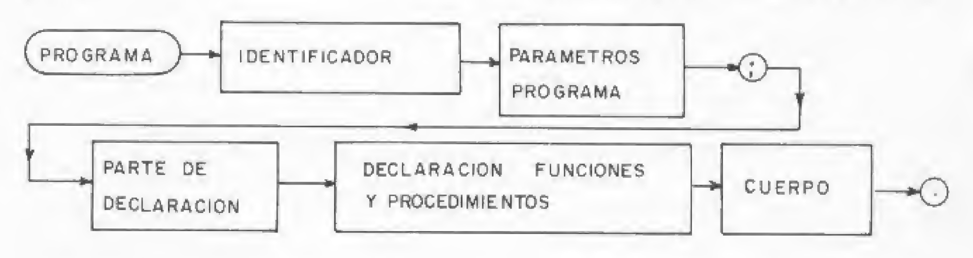
Las únicas palabras obligatorias en un programa son: PROGRAM, BEGIN y END. Las demás dependen de cada programa. Algunos compiladores exigen la declaración de variables obligatoriamente.

4. Desarrollo de un programa

Sabemos que el diagrama sintáctico de un identificador es



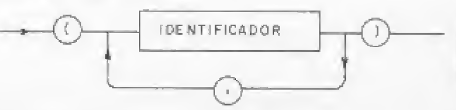
A partir de él vamos a definir la sintaxis de un programa PASCAL.



Vamos a explicar el diagrama anterior:

— Identificador: Será el nombre que le demos al programa.

— Parámetros del programa: Su diagrama es



PASCAL

Normalmente los parámetros de un programa son INPUT para lectora y OUTPUT para impresora.

— Parte de declaración: En ella se hará la declaración de las etiquetas, constantes, tipos y variables según este orden.

Veamos el diagrama general:

LABEL
ETIQUETA
;
,
CONST
IDENTIFICADOR
=
CONSTANTE
;
TYPE
IDENTIFICADOR
=
TIPO
;
VAR
IDENTIFICADOR
:
TIPO
;
,

Examinemos ahora cada una en particular:

a) LABEL

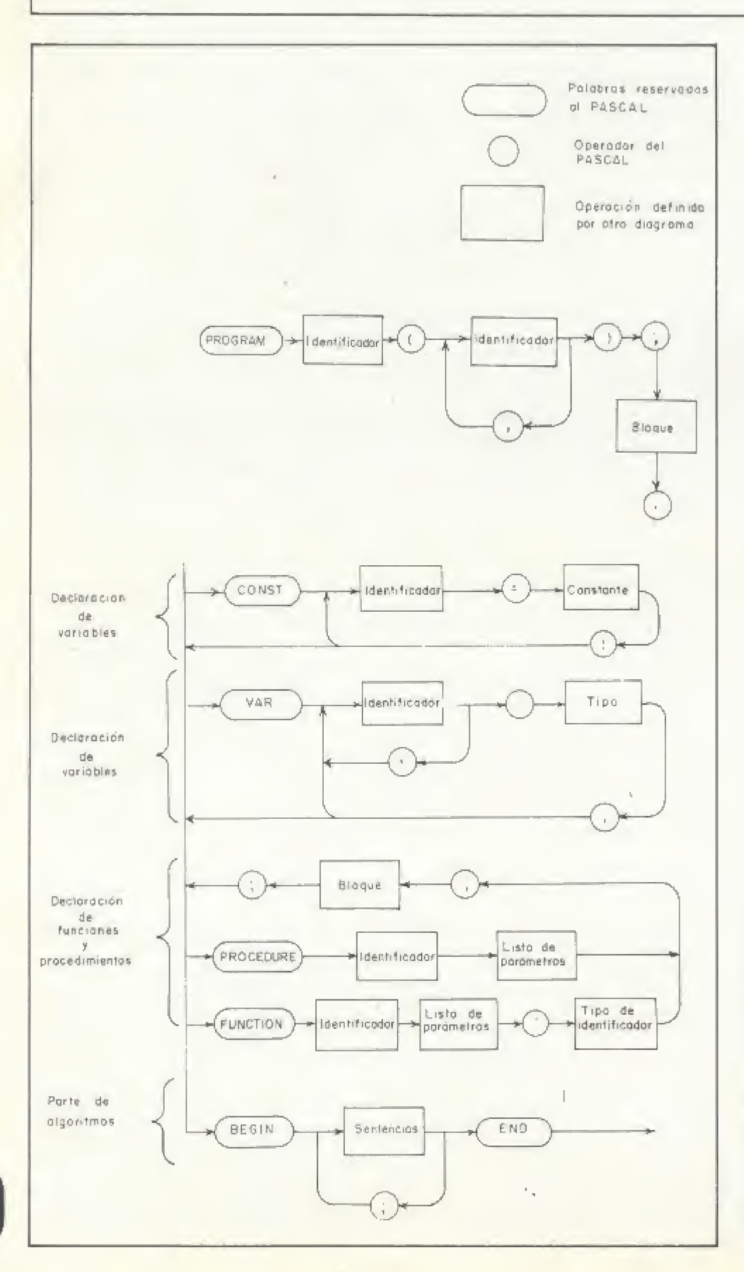
Para declarar las etiquetas pondremos en primer lugar la palabra reservada LABEL, y a continuación la lista de etiquetas separadas por comas. Las etiquetas serán números enteros entre 1 y 9999. Por ejemplo:

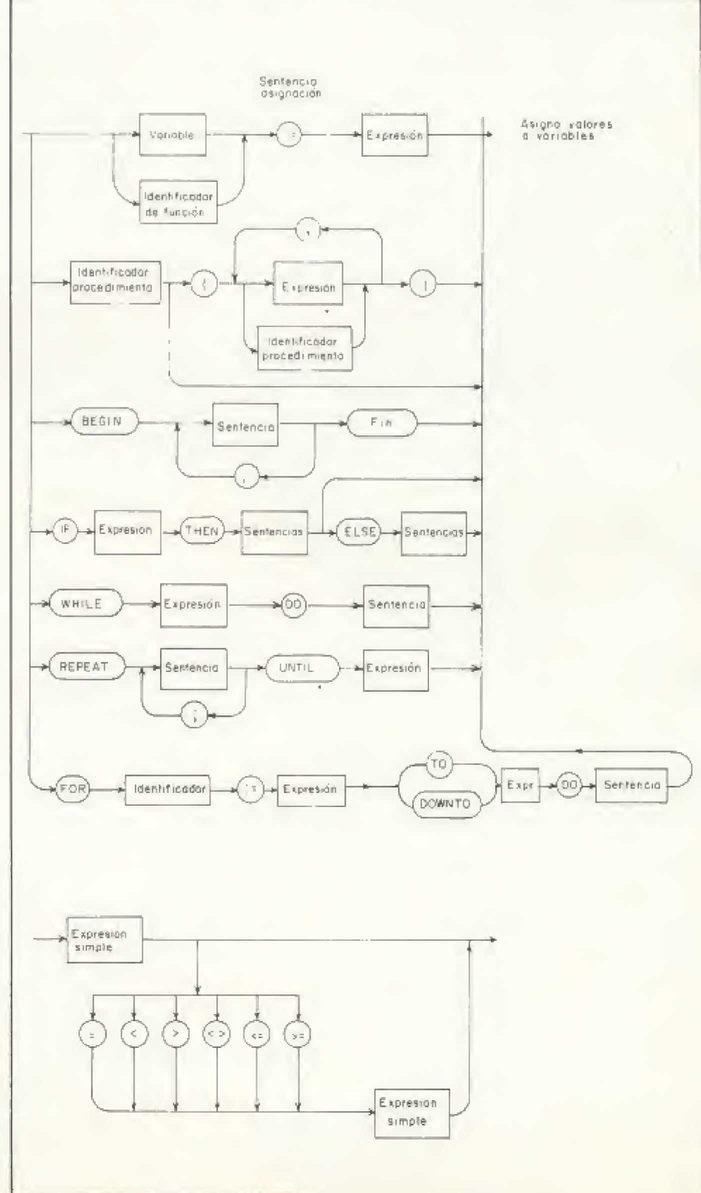
LABEL 957, 1, 32

La utilización de las etiquetas se produce en la sentencia de bifurcación GO TO, pero la utilizaremos muy poco ya que esta instrucción se trata de evitar lo más posible en programación estructurada.

b) CONST

Son datos que no varían a lo largo de la ejecución del programa.






ESPECIFICAMENTE PARA QL
Mod. 1431 MZ 4
P.V.P. 74.500 Ptas.
INMEDIATAMENTE DISPONIBLE
ESPECIFICACIONES
MODELO:
CUB 1451/DQ3
14" QL MONITOR
Entrada RGB-TTL
Resolución (PIXELS)
653 (H) × 585 (V)
DOT PITCH 0,43 mm
Bandwidth 18 MHz
Específicamente diseñado para el QL que aprovecha su facilidad única de 85 columnas.
Mod. 1451/DQ3
P.V.P. 96.000 Ptas.
MICROVITEC 653
cub
COLOUR DISPLAYS
DISTRIBUIDO EN EXCLUSIVA POR: MULTILOGIC. Ramón de Santillán, n.º 15 28016 MADRID Tel. 458 74 75

PASCAL

Ejemplos de declaración son:

CONST
pi = 3,14159;
diasemana = 7;

c) TYPE

Lo veremos más adelante.

d) VAR

Son datos que varían a lo largo del programa. Hay cinco tipos básicos:

— Enteras: INTEGER. Para enteros positivos y negativos.

— Reales: REAL. Para números reales.

— Booleano: BOOLEAN. Para variables de tipo lógico que pueden tomar dos valores TRUE o FALSE según se cumpla o no.

— Carácter: CHAR. Contienen caracteres alfabéticos y numéricos y además el carácter especial blanco.

— Matriz: ARRAY. Hablaremos de ello más adelante.

Ejemplos de declaración de variables son:

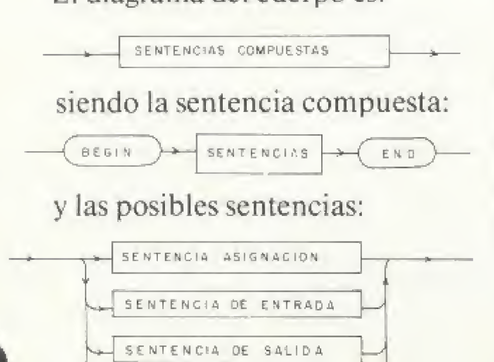
VAR
n.º alumnos aula: INTEGER;
madrileño: BOOLEAN;
peso: REAL;
horario: CHAR;

Aparte de estos cinco tipos básicos las variables podrán ser de los tipos definidos mediante la declaración TYPE.

La declaración de funciones y procedimientos la dejaremos para cuando hayamos avanzado un poco más. Lo único que indicaremos es que tienen una función parecida a las subrutinas en BASIC.

Vamos a explicar por último el cuerpo del programa.

El diagrama del cuerpo es:



siendo la sentencia compuesta:

y las posibles sentencias:

Antes de explicar el primer programa, vamos a explicar las sentencias fundamentales:

a) Sentencia de asignación

Es una de las más importantes y a la vez más sencilla.

Su diagrama es:



El signo := conviene leerlo como «se convierte en».

Esta instrucción ejecuta la expresión y asigna el valor de la ejecución a la variable. Sea por ejemplo:

Producto:= N1 * N2

si, por ejemplo, N1 contiene un 5, y N2 un 2, primero se hará el producto 5 * 2 = 10, y se almacenará en Producto, luego la variable producto contendrá desde ese momento el valor 10.

b) Sentencia de Entrada

La entrada standard se hará mediante la instrucción READ, que permite la lectura de un dato y su asignación a una variable del programa. Su formato es: READ,

Tipo simple — ARRAY [Tipo simple ,] OF Tipo

Tipo identificador — (Identificador ,) — Constante .. Constante

(Identificador , : Tipo identificador ;) — VAR

Digito

Letra — Letra — Digito

(dato1, dato2, ...); también existe la opción READLN (abreviatura de READ LINE) en la cual, cada vez que se lee un dato se salta de línea para leer el siguiente en la siguiente línea. Su formato es: READLN (dato1, dato2, ...); veamos un ejemplo para la aclaración de estas sentencias:

Si la entrada de datos es:

```
25    4
84    23
```

La sentencia READ (A,B) asignará a la variable A el valor 25, y a la variable B el valor 4. Por otra parte, la sentencia READLN (A,B), asignará a la variable A el valor 25, y a la variable B el valor 84.

c) Sentencia de salida

La salida standard se hará mediante la instrucción WRITE. Se utiliza igual que la instrucción PRINT en BASIC. Sólo permite la escritura de valores numéricos de las variables. Si queremos incluir en la sentencia un literal, éste deberá ir encerrado entre comilla ('). Su formato es WRITE (resultado1, resultado2, ...); si queremos escribir por líneas usaremos la opción WRITELN.

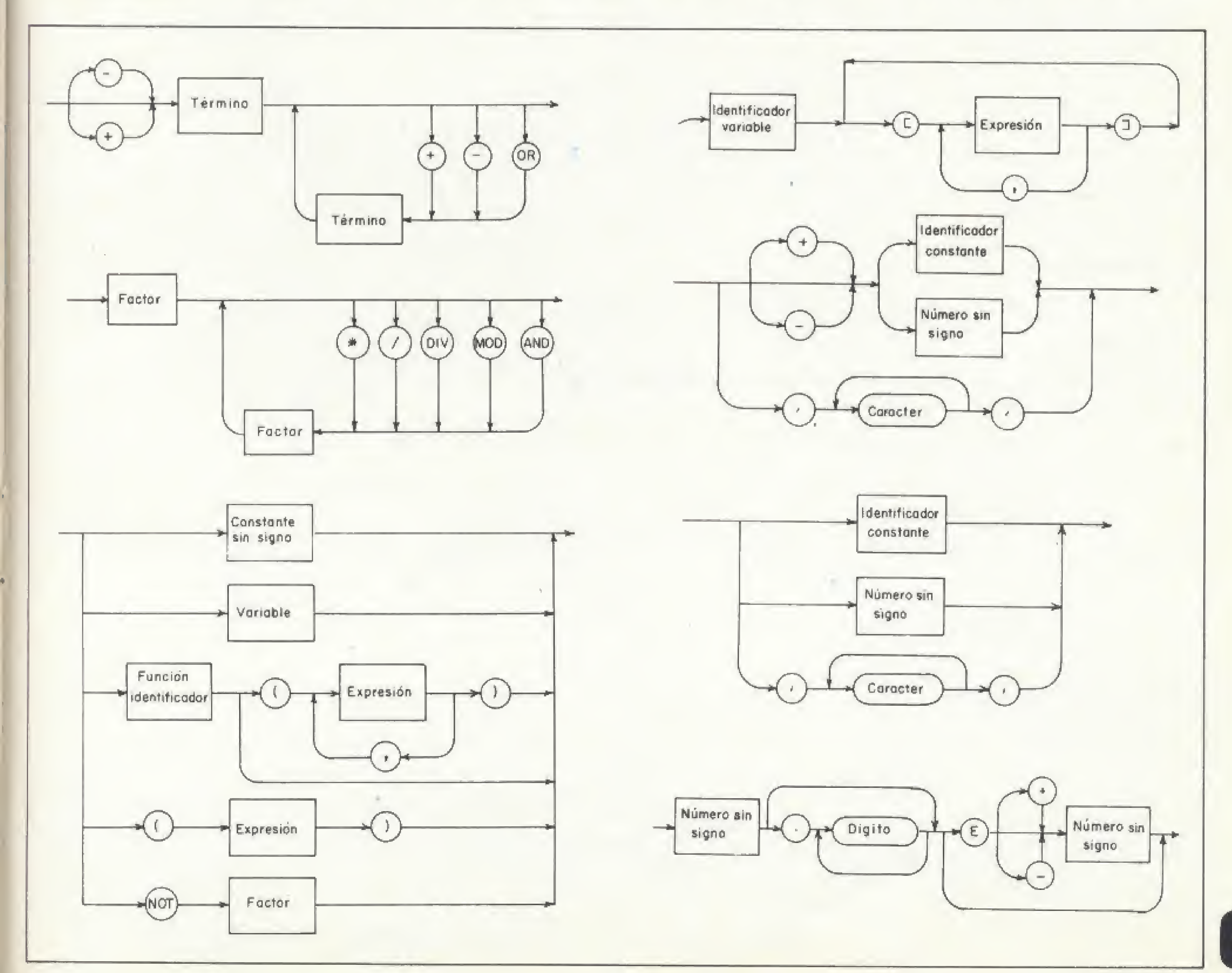
Vamos a ver la diferencia entre WRITE y WRITELN mediante dos pequeños programas. Prueba estos programas y observa la diferencia en las salidas:

```
PROGRAM Saludo;
BEGIN
WRITE ('Esto se escribirá en
una línea')
END.
PROGRAM Saludo2;
BEGIN
WRITELN ('Esto se escribirá');
WRITELN ('En dos líneas')
END.
```

La utilización de WRITELN sin parámetros producirá una línea en blanco. Prueba a incluir otra instrucción WRITELN en medio de las dos sentencias de escritura de Saludo2.

Por último existe otra opción en la sentencia de escritura, y es indicarle al ordenador la limitación de caracteres de la salida.

Su formato es: WRITE (resultado : e : e_1);
siendo «e» el n.º de caracteres con que queremos escribir el resultado. Cuando vayamos a escribir un número decimal, podemos indicar también mediante la opción e_1 cuántos decimales queremos que salgan escritos, es decir, de los «e» caracteres que se escribirán, «e_1» estarán detrás del punto decimal.





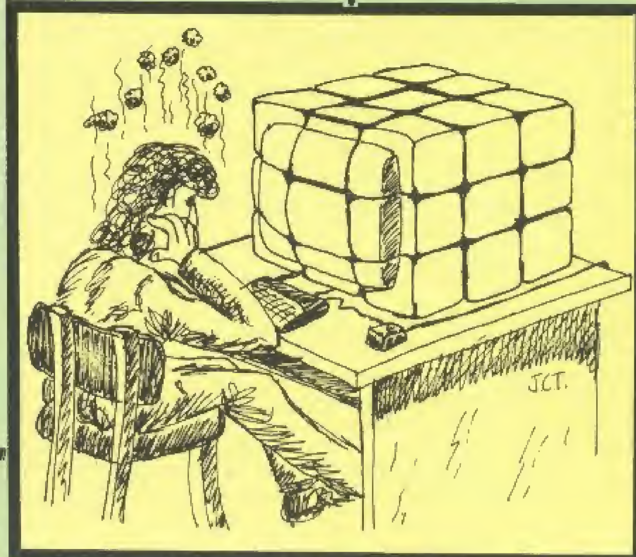
Para estas navidades no se lo vamos a poner muy difícil. Empezamos con dos dimensiones... pero le prometemos las cuatro para un próximo número.

Nueve niveles de dificultad le permiten elegir la modalidad de juego más acorde con la complejidad de sus neuronas. Y no lo tome a broma, para salir airoso del nivel nueve necesitará bastante habilidad y no menos destreza.

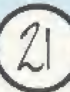
Elegido el nivel aparecerá en la pantalla un cuadro de 400 casillas (20 × 20). El ordenador comenzará a cambiar el modo de impresión de las casillas aleatoriamente: normal, inverso, normal, etc. A mayor nivel elegido, mayor será el número de inversiones realizadas (la combinación resultante después de elegir el nivel 9, no es apta para cardíacos).

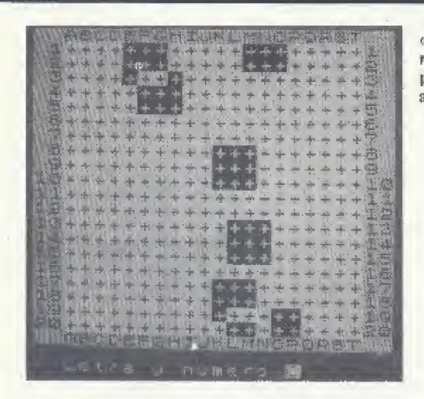
Para ganar el juego, ha de lograr que todas las casillas estén en blanco, es decir, que todas estén en modo normal y ninguna en inverso. Para ello, en cada jugada ha de dar la posición de una casilla referenciada por sus coordenadas, primero la columna y después la fila. Comprobará que esta casilla, y todas las que se encuentran a su alrededor, cambian del modo normal al inverso o al contrario, según la situación en que se encontrasen. ¿Ha comprendido el funcionamiento? Le aseguramos que es posible ganar, pero no le decimos en el nivel que lo logramos.

Programas



EL CUBO DE RUBIK





«Ejemplo de la representación en pantalla, jugando a nivel 2».

```
   5 POKE 23658,8: REM Mayuscul
as.
   6 LET goes=0
   7 PAPER 1: BORDER 1: INK 7: C
LS
   8 GO TO 22
  10 CLS : FOR f=0 TO 7: PRINT
PAPER f; INK 9;TAB f;"PUZZLE": N
EXT f
  20 FOR f=7 TO 0 STEP -1: PRINT
  PAPER f; INK 9;TAB f;"PUZZLE":
 NEXT f
  21 RETURN
  22 GO SUB 10: PRINT AT 20,0;"I
nstrucciones s/n."
  24 IF INKEY$<>"S" AND INKEY$<>
"N" THEN  GO TO 24
  26 IF INKEY$="S" THEN  GO SUB
6000: REM instruccciones
  30 GO SUB 10: PRINT #1;"Dificu
ltad (1 TO 9) ?"
  35 LET d$=INKEY$: IF d$="" THE
N  GO TO 35
  40 IF d$<"1" OR d$>"9" THEN  G
O TO 35
  45 LET di=INT VAL d$
  50 LET di=di*10
  70 GO SUB 1000: REM dibujar ta
blero.
  80 LET nor=400: LET inv=0
  90 GO SUB 3000: REM Manejo tab
lero.
 100 REM Tablero.
 105 LET goes=goes+1
 107 PRINT #1;AT 0,0;"Movimiento
 ";goes
 110 INPUT AT 1,1;" Letra y nume
ro "; LINE m$: IF LEN m$>3 OR LE
N M$<2 THEN  GO TO 110
 120 LET l$=m$(1): LET m$=m$(2 T
O ): IF l$<"A" OR l$>"T" THEN  G
O TO 110
 130 IF m$<"1" OR m$>"9" THEN  G
O TO 110
 140 LET x=CODE l$-63: LET y=VAL
 m$
 150 GO SUB 2000: REM cambiar pa
ntalla.
 160 IF NOR=400 OR INV=400 THEN
 GO TO 5000: REM Completado.
 170 GO TO 100
1000 REM Pantalla.
1001 CLS : INK 0
1005 PRINT  PAPER 6;"  ABCDEFGHI
JKLMNOPQRST  "
1010 FOR f=1 TO 20
1015 IF f<10 THEN  PRINT  PAPER
6;" ";
1020 PRINT  PAPER 6;f; PAPER 7;"
++++++++++++++++++++"; PAPER 6;f
;: IF F<10 THEN  PRINT  PAPER 6;
" ";
1025 PRINT
1030 NEXT f
1040 PRINT  PAPER 6;"  ABCDEFGHI
JKLMNOPQRST  "
1045 INK 7
1050 RETURN
2000 REM Cambiar pantalla.
2020 IF ATTR (y,x)=56 THEN  PRIN
T AT y,x; INK 7; PAPER 0;"+": LE
T inv=inv+1: LET nor=nor-1: GO T
O 2030
2025 IF ATTR (y,x)<>48 THEN  PRI
NT AT y,x; INK 0; PAPER 7;"+": L
ET nor=nor+1: LET inv=inv-1
2030 IF ATTR (y,x+1)=56 THEN  PR
INT AT y,x+1; INK 7; PAPER 0;"+"
: LET inv=inv+1: LET nor=nor-1:
GO TO 2040
2035 IF ATTR (y,x+1)<>48 THEN  P
RINT AT y,x+1; INK 0; PAPER 7;"+
": LET nor=nor+1: LET inv=inv-1
2040 IF ATTR (y,x-1)=56 THEN  PR
INT AT y,x-1; INK 7; PAPER 0;"+"
: LET inv=inv+1: LET nor=nor-1:
GO TO 2050
2045 IF ATTR (y,x-1)<>48 THEN  P
RINT AT y,x-1; INK 0; PAPER 7;"+
": LET nor=nor+1: LET inv=inv-1
2050 IF ATTR (y+1,x)=56 THEN  PR
INT AT y+1,x; INK 7; PAPER 0;"+"
: LET inv=inv+1: LET nor=nor-1:
GO TO 2060
2055 IF ATTR (y+1,x)<>48 THEN  P
RINT AT y+1,x; INK 0; PAPER 7;"+
": LET nor=nor+1: LET inv=inv-1
2060 IF ATTR (y-1,x)=56 THEN  PR
INT AT y-1,x; INK 7; PAPER 0;"+"
: LET inv=inv+1: LET nor=nor-1:
```

```
GO TO 2070
2065 IF ATTR (y-1,x)<>48 THEN  P
RINT AT y-1,x; INK 0; PAPER 7;"+
": LET nor=nor+1: LET inv=inv-1
2070 IF ATTR (y-1,x-1)=56 THEN
PRINT AT y-1,x-1; INK 7; PAPER 0
;"+": LET inv=inv+1: LET nor=nor
-1: GO TO 2080
2075 IF ATTR (y-1,x-1)<>48 THEN
 PRINT AT y-1,x-1; INK 0; PAPER
7;"+": LET nor=nor+1: LET inv=in
v-1
2080 IF ATTR (y+1,x-1)=56 THEN
PRINT AT y+1,x-1; INK 7; PAPER 0
;"+": LET inv=inv+1: LET nor=nor
-1: GO TO 2090
2085 IF ATTR (y+1,x-1)<>48 THEN
 PRINT AT y+1,x-1; INK 0; PAPER
7;"+": LET nor=nor+1: LET inv=in
v-1
2090 IF ATTR (y+1,x+1)=56 THEN
PRINT AT y+1,x+1; INK 7; PAPER 0
;"+": LET inv=inv+1: LET nor=nor
-1: GO TO 2100
2095 IF ATTR (y+1,x+1)<>48 THEN
 PRINT AT y+1,x+1; INK 0; PAPER
7;"+": LET nor=nor+1: LET inv=in
v-1
2100 IF ATTR (y-1,x+1)=56 THEN
PRINT AT y-1,x+1; INK 7; PAPER 0
;"+": LET inv=inv+1: LET nor=nor
-1: GO TO 2110
2105 IF ATTR (y-1,x+1)<>48 THEN
 PRINT AT y-1,x+1; INK 0; PAPER
7;"+": LET nor=nor+1: LET inv=in
v-1
2110 RETURN
3000 REM Jugada.
3010 FOR f=1 TO di
3020 LET x=INT (RND*20)+2: LET y
=INT (RND*20)+1
3030 GO SUB 2000
3040 NEXT f
3050 RETURN
5000 REM Completado
5010 FOR f=0 TO 30 STEP .3
5020 BEEP .003,f: BEEP .003,f+10
5030 NEXT f
5040 CLS : PRINT AT 10,3; INK 7;
 PAPER 5; FLASH 1; BRIGHT 1;"Exc
elente!!": PRINT AT 15,0;"Echamo
s otra?"
5045 PRINT AT 12,4;"Necesitaste
"; FLASH 1;goes; FLASH 0;" inten
tos."
5050 IF INKEY$="" THEN  GO TO 50
50
5060 IF INKEY$="S" THEN  GO TO 1
5070 STOP
5999 REM instrucciones
6000 CLS
6010 PRINT TAB 9;"**************
"
6020 PRINT TAB 9;"*   PUZZLE   *
"
6030 PRINT TAB 9;"**************
"
6040 PRINT '"PUZZLE es una versi
on del cubo  de Rubic en dos dim
ensiones. El tablero es de 20*20
. El Spec-   trum seleccionara d
istintas po- siciones aleatoriam
ente y la in-vierte junto con la
s siete casi-llas vecinas. Este
proceso se   repite una serie de
 veces segun el nivel de dificul
tad."
6050 PRINT '"Su unico ";
6060 FOR i=1 TO 5: PRINT CHR$ 8;
: NEXT i
6070 PRINT  OVER 1;"______"; OVE
R 0;" trabajo es dejar el ta-ble
ro en su estado original, esdeci
r, blanco."
6080 PRINT AT 20,5; FLASH 1;"Pre
sione cualquier tecla para   con
tinuar"
6090 IF INKEY$="" THEN  BEEP .1,
10: GO TO 6090
6100 FOR i=1 TO 20: BEEP .1,i: N
EXT i
6110 RETURN
```

Leyendo entre líneas

Líneas 30-50	Introducción nivel de dificultad. La variable «d» se multiplica por 10 y el valor resultante es el número mínimo de intentos para ganar el juego.
Líneas 70-90	Dibujo de las casillas (Subrutina 1000) y realización de inversiones según el nivel elegido (Subrutina 3000).
Líneas 100-170	Bucle principal. Se incrementa el contador de jugadas, entrada de datos y verificación de su validez. La línea 140 calcula la posición de la casilla y la 160 comprueba si el juego ha finalizado.
Líneas 1000-1050	Rutina de impresión de las casillas iniciales.
Líneas 2000-2105	Rutina de cambio de la casilla elegida y las adyacentes.
Líneas 2110-3050	Rutina de inversión aleatoria utilizada al inicio del juego. El número de inversiones viene determinado por la variable «di», de nivel de juego.

Parejas

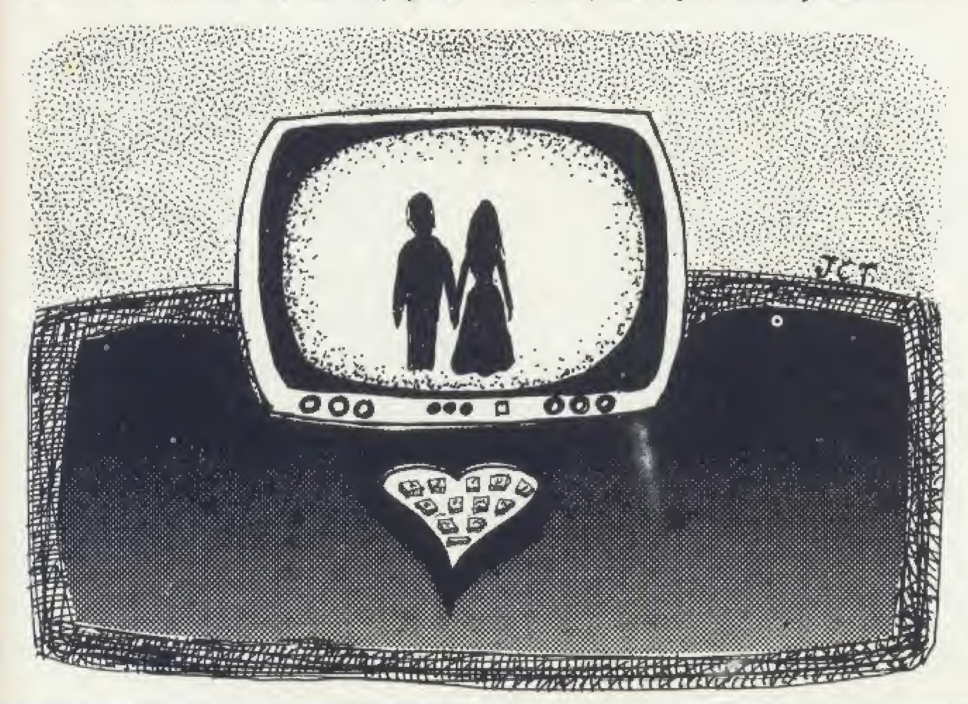
Ultimamente las agencias matrimoniales parecen estar de moda. Muchas de ellas quizá trabajan con un programa similar al que ofrecemos a continuación, que sin embargo no fue ideado para "buscar" al compañero ideal, sino para pasar un buen rato. Así nos lo decía su autor, Carlos Quintana, director de *marketing* de IBM en Barcelona: "Con cierta frecuencia nos reunimos un grupo de amigos a cenar y quería enseñarles mi nuevo ordenador con algún programa de cierta gracia. Así nació este programa, con las sugerencias de mi mujer y mi cuñada. A veces lo probamos con una o dos parejas, pero no tiene gracia. El límite es de diez por la presentación en pantalla, pero más de ocho resulta un poco lento. Puedo asegurar que funciona, ya que generalmente salen personas compatibles, aunque no siempre concuerda con los matrimonios".

Carlos Quintana, director de marketing de IBM en Barcelona.



No pudimos resistir la tentación de preguntarle el por qué un vendedor de IBM habia decidido comprado el Sinclair, sobre para el Spectrum. "No he sido sólo yo. Compañeros míos en IBM se han comprado el Sinclair, sobre todo para meterse dentro de la maquinaria y ver sus posibilidades. Yo aprendí a programar en Cobol,

Assembler y RP6, lenguajes que entonces estaban de moda, pero fue hacer el curso de la empresa de 15 días, unas prácticas y listos. Lo compré por los chicos y el por qué Sinclair se debe al bajo precio".

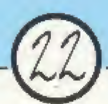


Las instrucciones REM (comentarios), están en catalán. "Lo tenía hecho en catalán, y lo pasé al castellano para su publicación... pero olvidé los comentarios".

Entrando en la descripción del programa, clasifica y empareja de acuerdo con las variables descritas en las líneas 60 a 200 según su ponderación de importancia de 1 a 5, tal y como se explican por pantalla. Obviamente, puede ampliar o cambiar el número de variables.

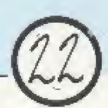
Reúnase con sus amigos y adivine si su chico(a) favorito(a) es realmente quien más le conviene. Quizá muy cerca de usted está su "media naranja"... ¡Y usted sin enterarse!

Autor: Carlos Quintana *48K*



```
  30 LET c=15
  40 LET ll=0
  41 PRINT AT 5,12;"PAREJAS";AT
6,12;"        ": PRINT AT
10,10;"C.Quintana"
  50 DIM c$(c,14)
  60 LET c$(1)="LECTURA"
  70 LET c$(2)="VIDA SOCIAL"
  80 LET c$(3)="VIDA FAMILIAR"
  90 LET c$(4)="DEPORTE"
 100 LET c$(5)="BUENA COMIDA"
 110 LET c$(6)="VIDA SEXUAL"
 120 LET c$(7)="VIAJES"
 130 LET c$(8)="RELIGION"
 140 LET c$(9)="JUERGA"
 150 LET c$(10)="BAILE"
 160 LET c$(11)="TRABAJO"
 170 LET c$(12)="AHORRO"
 180 LET c$(13)="ORDENADO"
 190 LET c$(14)="DETALLISTA"
 200 LET c$(15)="TELEVISION"
 210 DIM c(15)
 220 LET c(1)=0
 230 LET c(2)=1
 240 LET c(3)=1
 250 LET c(4)=0
 260 LET c(5)=0
 270 LET c(6)=2
 280 LET c(7)=1
 290 LET c(8)=2
 300 LET c(9)=1
 310 LET c(10)=0
 320 LET c(11)=2
 330 LET c(12)=2
 340 LET c(13)=1
 350 LET c(14)=1
 360 LET c(15)=0
 370 PAUSE 200: CLS
 380 PRINT AT 5,1;"ENTRAR NUMERO
 DE PARTICIPANTES"
 390 PRINT AT 8,3;"HOMBRES = ?"
 400 GO SUB 3330
 410 LET h=VAL w$
 420 IF h>9 THEN  PRINT AT 10,3;
 FLASH 1;"MAXIMO 9": BEEP 1,-11:
 PAUSE 30: PRINT AT 10,3;"
   ": GO TO 400
 430 PRINT AT 8,13;h
 440 DIM h$(h,12)
 450 PRINT AT 8,18;"MUJERES = ?"
 460 GO SUB 3330
 470 LET m=VAL w$
 480 IF m>9 THEN  PRINT AT 10,18
; FLASH 1;"MAXIMO 9": BEEP 1,-11
: PAUSE 40: PRINT AT 10,18;"
     ": GO TO 460
 490 PRINT AT 8,28;m
 500 DIM m$(m,12)
 510 PAUSE 100
 520 CLS
 530 PRINT AT 3,1;"NOMBRES DE LO
S HOMBRES"
 540 FOR x=1 TO h
 550 PRINT AT 5+x,5;x;" = ?"
 560 INPUT  LINE w$
 570 GO SUB 3400
 580 LET h$(x)=w$
 590 IF x=1 THEN  GO TO 650
 600 FOR w=1 TO x-1
 610 IF h$(w)<>h$(x) THEN  GO TO
 640
 620 BEEP 1,-11: PRINT AT 5+x,9;
 FLASH 1;"NOMBRE REPETIDO"
 630 GO TO 560
 640 NEXT w
 650 PRINT AT 5+x,9;h$(x)
 660 NEXT x
 670 PAUSE 100
 680 CLS
 690 PRINT AT 3,1;"NOMBRES DE LA
S MUJERES"
 700 FOR x=1 TO m
 710 PRINT AT 5+x,5;x;" = ?"
 720 INPUT  LINE w$
 730 GO SUB 3400
 740 LET m$(x)=w$
 750 IF x=1 THEN  GO TO 810
 760 FOR w=1 TO x-1
 770 IF m$(w)<>m$(x) THEN  GO TO
 800
 780 BEEP 1,-11: PRINT AT 5+x,9;
 FLASH 1;"NOMBRE REPETIDO"
 790 GO TO 720
 800 NEXT w
 810 PRINT AT 5+x,9;m$(x)
 820 NEXT x
 830 PAUSE 100
 840 PRINT AT 3,1;"CARACTERISTIC
AS A VALORAR"
 850 FOR x=1 TO c
 860 PRINT AT 4+x,5-(x>9);x;".-"
;c$(x)
 870 NEXT x
 880 PRINT AT 21,1;"ENTRA -1- PA
RA CONTINUAR"
 890 INPUT  LINE w$
```

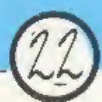
```
 900 CLS
 910 PRINT AT 0,0;"PARA CADA CAR
ACTERISTICA INDICAR"
 920 PRINT AT 6,4;"1 = INTERES N
ULO";AT 8,4;"2 = POCO INTERES";A
T 10,4;"3 = NORMAL";AT 12,4;"4 =
 BASTANTE INTERES";AT 14,4;"5 =
MUCHO INTERES"
 930 PRINT AT 21,1;"ENTRA -1- PA
RA CONTINUAR"
 940 INPUT  LINE w$
 950 CLS
 960 DIM b(h,c)
 970 FOR x=1 TO h
 980 PRINT AT 3,3;"PREFERENCIAS
DE ";h$(x)
 990 FOR y=1 TO c
1000 PRINT AT 4+y,5;c$(y);" = ?"
1010 GO SUB 3270
1020 LET b(x,y)=VAL w$
1030 PRINT AT 4+y,22;b(x,y)
1040 NEXT y
1050 CLS
1060 NEXT x
1070 CLS
1080 DIM v(m,c)
1090 FOR x=1 TO m
1100 PRINT AT 3,3;"PREFERENCIAS
DE ";m$(x)
1110 FOR y=1 TO c
1120 PRINT AT 4+y,5;c$(y);" = ?"
1130 GO SUB 3270
1140 LET v(x,y)=VAL w$
1150 PRINT AT 4+y,22;v(x,y)
1160 PRINT AT 4+y,22;v(x,y)
1170 NEXT y
1180 CLS
1190 NEXT x
1200 DIM d(h,m)
1210 DIM i(m,h)
1220 DIM t(h,m)
1230 LET tc=343.56
1240 FOR x=1 TO h
1250 PRINT AT 2,3;x;"/";h
```



```
1260 FOR y=1 TO m
1270 PRINT AT 2,12;y;"/";m
1280 FOR z=1 TO c
1290 LET d(x,y)=d(x,y)+((b(x,z)-
v(y,z))*(1+0.2*c(z)))*((b(x,z)-v
(y,z))*(1+0.2*c(z)))
1300 NEXT z
1310 LET d(x,y)=INT d(x,y)+y/10
1320 LET i(y,x)=INT d(x,y)+x/10
1330 LET t(x,y)=INT (((tc-d(x,y)
)/tc)*100)
1340 NEXT y
1350 NEXT x
1360 DIM p(m*h)
1370 LET k=1
1380 FOR x=1 TO h
1390 FOR y=1 TO m
1400 LET p(k)=d(x,y)+x/100
1410 LET k=k+1
1420 NEXT y
1430 NEXT x
1440 CLS
1450 FOR z=1 TO h
1460 FOR x=1 TO m-1
1470 FOR y=x+1 TO m
1480 IF d(z,x)<d(z,y) THEN  GO T
O 1520
1490 LET e=d(z,x)
1500 LET d(z,x)=d(z,y)
1510 LET d(z,y)=e
1520 NEXT y
1530 NEXT x
1540 NEXT z
1550 FOR z=1 TO m
1560 FOR x=1 TO h-1
1570 FOR y=x+1 TO h
1580 IF i(z,x)<i(z,y) THEN  GO T
O 1620
1590 LET e=i(z,x)
1600 LET i(z,x)=i(z,y)
1610 LET i(z,y)=e
1620 NEXT y
1630 NEXT x
1640 NEXT z
1650 CLS
1660 PRINT AT 3,1;"ESCOGER ENTRE
 :"
1670 PRINT AT 6,4;"1. PREFERENCI
AS DE CADA UNO";AT 8,4;"2. VER L
AS DE UN HOMBRE";AT 10,4;"3. VER
 LAS DE UNA MUJER"
1680 PRINT AT 12,4;"4. SELECCION
 DE PAREJAS"
1690 PRINT AT 14,4;"5. CLASIFICA
CION DE HOMBRES";AT 16,4;"6. CLA
SIFICACION DE MUJERES"
1700 INPUT  LINE w$
1710 IF w$<"1" OR w$>"6" THEN  G
O TO 1700
1720 LET r=VAL w$
1730 CLS
1740 IF r=1 THEN  GO TO 1800
1750 IF r=2 THEN  GO TO 1870
1760 IF r=3 THEN  GO TO 1940
1770 IF r=4 THEN  GO TO 2010
1780 IF r=5 THEN  GO TO 2030
1790 GO TO 2040
1800 FOR x=1 TO h
1810 GO SUB 2050
1820 NEXT x
1830 FOR x=1 TO m
1840 GO SUB 2130
1850 NEXT x
1860 GO TO 2210
1870 CLS
1880 PRINT AT 3,3;"NUMERO DEL HO
MBRE = ?"
1890 INPUT  LINE w$
1900 IF w$<"1" OR w$>STR$ h THEN
  GO TO 1890
1910 LET x=VAL w$
1920 GO SUB 2050
1930 GO TO 1660
1940 CLS
1950 PRINT AT 3,3;"NUMERO DE LA
MUJER = ?"
1960 INPUT  LINE w$
1970 IF w$<"1" OR w$>STR$ m THEN
  GO TO 1960
1980 LET x=VAL w$
1990 GO SUB 2130
2000 GO TO 1660
2010 CLS
2020 GO TO 2210
2030 GO TO 2670
2040 GO TO 2910
2050 CLS
2060 PRINT AT 3,1;"PARA ";h$(x);
AT 5,1;"LAS PREFERIDAS SON :"
2070 FOR y=1 TO m
2080 PRINT AT 6+y,3;y;". ";m$((d
(x,y)-INT d(x,y))*10);TAB 19;t(x
,(d(x,y)-INT d(x,y))*10);TAB 22;
"%"
2090 NEXT y
2100 GO SUB 3170
```



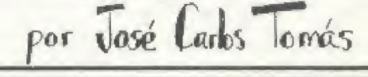
bip-bip
P.V.P. 500 Ptas.
EL AMIGO DE TU ORDENADOR
Programas y juegos para tu micro
A
Diccionario completo de informática
Ordenadores:
Cómo funcionan, cual comprar, para que sirven
YA ESTA A LA VENTA
«YO SOY BIP-BIP. Y QUIERO QUE SEAMOS AMIGOS»
Comics y cuentos
Aprende a programar

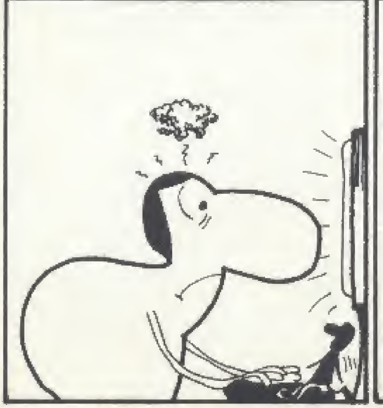


```
2110 CLS
2120 RETURN
2130 CLS
2140 PRINT AT 3,1;"PARA ";m$(x);
AT 5,1;"LOS PREFERIDOS SON :"
2150 FOR y=1 TO h
2160 PRINT AT 6+y,3;y;" . ";h$((
i(x,y)-INT i(x,y))*10);TAB 19;t(
(i(x,y)-INT i(x,y))*10,x);TAB 22
;"%"
2170 NEXT y
2180 GO SUB 3170
2190 CLS
2200 RETURN
2210 PRINT AT 3,5;"PAREJAS SELEC
CIONADAS"
2220 FOR x=1 TO m*h-1
2230 PRINT AT 10,12;x;"/";m*h-1
2240 FOR y=x+1 TO m*h
2250 IF p(x)<p(y) THEN  GO TO 22
90
2260 LET e=p(x)
2270 LET p(x)=p(y)
2280 LET p(y)=e
2290 NEXT y
2300 NEXT x
2310 DIM a(h)
2320 PRINT AT 4,5;"
                  "
2330 DIM g(m)
2340 LET s=m
2350 IF h<m THEN  LET s=h
2360 LET y=1
2370 FOR k=1 TO s
2380 FOR x=y TO m*h
2390 PRINT AT 14,12;x;"/";m*h-1
2400 IF k=1 THEN  GO TO 2470
2410 FOR l=1 TO k-1
2420 IF INT (10*(p(x)-INT p(x)))
=g(l) THEN  NEXT x
2430 NEXT l
2440 FOR z=1 TO k-1
2450 IF INT (0.0001+100*p(x)-10*
INT (10*p(x)))=a(z) THEN  NEXT x
2460 NEXT z
2470 LET g(k)=INT (10*(p(x)-INT
p(x)))
2480 LET a(k)=INT (0.0001+(100*p
(x))-10*INT (10*p(x)))
2490 LET y=x+1
2500 NEXT k
2510 CLS
2520 PRINT AT 3,5;"PAREJAS SELEC
CIONADAS";AT 4,5;"
                  "
2530 FOR k=1 TO s
2540 PRINT AT 5+2*k,1;h$(a(k));m
$(g(k));t(a(k),g(k));TAB 28;"%"
2550 NEXT k
2560 IF h=m THEN  GO TO 2660
2570 PRINT AT 7+2*k,0;"QUEDAN SI
N APAREJAR:"
2580 IF h<m THEN  GO TO 2630
2590 FOR x=s+1 TO h
2600 PRINT AT 8+2*k+x-s,21;h$(IN
T (0.0001+(100*p(x))-10*INT (10*
p(x))))
2610 NEXT x
2620 GO TO 2660
2630 FOR x=s+1 TO m
2640 PRINT AT 8+2*k+x-s,21;m$(IN
T (10*(p(x)-INT p(x))))
2650 NEXT x
2660 GO SUB 3170
2670 CLS
2680 DIM s(h)
2690 PRINT AT 1,2;"CLASIFICACION
 DE LOS HOMBRES";AT 3,4;"POR SU
ACEPTACION ENTRE";AT 5,10;"LAS M
UJERES"
2700 FOR x=1 TO h
2710 PRINT AT 10,14;x;"/";h
2720 FOR y=1 TO m
2730 LET s(x)=s(x)+t(x,y)
2740 NEXT y
2750 LET s(x)=INT s(x)+x/10
2760 NEXT x
2770 FOR x=1 TO h-1
2780 FOR y=x+1 TO h
2790 IF s(x)>s(y) THEN  GO TO 28
30
2800 LET e=s(x)
2810 LET s(x)=s(y)
2820 LET s(y)=e
2830 NEXT y
2840 NEXT x
2850 DIM f(h)
2860 FOR x=1 TO h
2870 LET f(x)=10*(s(x)-INT s(x))
2880 PRINT AT 9+x,5;x;TAB 7;h$(f
(x));INT (s(x)/m);"%"
2890 NEXT x
2900 GO SUB 3170
2910 CLS
2920 DIM w(m)
2930 PRINT AT 1,2;"CLASIFICACION
```

```
 DE LAS MUJERES";AT 3,4;"POR SU
ACEPTACION ENTRE";AT 5,10;"LOS H
OMBRES"
2940 FOR x=1 TO m
2950 PRINT AT 10,14;x;"/";m
2960 FOR y=1 TO h
2970 LET w(x)=w(x)+t(y,x)
2980 NEXT y
2990 LET w(x)=INT w(x)+x/10
3000 NEXT x
3010 FOR x=1 TO m-1
3020 FOR y=x+1 TO m
3030 IF w(x)>w(y) THEN  GO TO 30
70
3040 LET e=w(x)
3050 LET w(x)=w(y)
3060 LET w(y)=e
3070 NEXT y
3080 NEXT x
3090 DIM j(m)
3100 FOR x=1 TO m
3110 LET j(x)=10*(w(x)-INT w(x))
3120 PRINT AT 9+x,5;x;TAB 7;m$(j
(x));INT (w(x)/h);"%"
3130 NEXT x
3140 LET ll=1
3150 GO SUB 3170
3160 GO TO 3450
3170 PRINT AT 20,0;"ENTRA-1-PARA
 CONTINUAR";AT 21,5;"-2-PARA VOL
VER A ESCOGER"
3180 IF ll=1 THEN  PRINT AT 20,1
3;"ACABAR    "
3190 LET ll=0
3200 INPUT  LINE w$
3210 IF w$<"1" OR w$>"2" THEN  G
O TO 3200
3220 LET mm=VAL w$
3230 CLS
3240 IF mm=2 THEN  GO TO 1660
3250 RETURN
3260 REM Entrada de valors 1 a 5
3270 INPUT  LINE w$
3280 IF w$="fi" THEN  GO TO 3450
3290 IF LEN w$>1 THEN  BEEP 1,-1
1: GO TO 3270
3300 IF w$<"1" OR w$>"5" THEN  B
EEP 1,-11: GO TO 3270
3310 RETURN
3320 REM Comprovacio del caracte
r numeric d'un numero.
3330 INPUT  LINE w$
3340 IF w$="fi" THEN  GO TO 3450
3350 FOR w=1 TO LEN w$
3360 IF w$(w)<"0" OR w$(w)>"9" T
HEN  BEEP 1,-11: GO TO 3330
3370 NEXT w
3380 RETURN
3390 REM Rutina de transformacio
 a majusculas
3400 FOR w=1 TO LEN w$
3410 IF w$(w)<"a" OR w$(w)>"z" T
HEN  GO TO 3430
3420 LET w$(w)=CHR$ ((CODE w$(w)
)-32)
3430 NEXT w
3440 RETURN
3450 PRINT AT 10,13;"ADIOS"
3460 STOP
9900 GO TO 1
```

GUSANEZ









23

Tenis



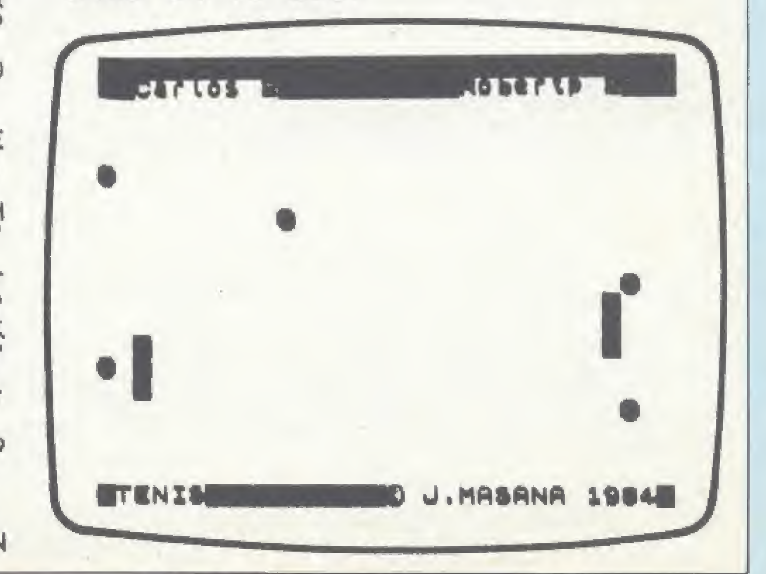
¿Se acuerda de aquellas primeras máquinas que irrumpieron súbitamente en todas las cafeterías, y con las que se realizaban auténticos torneos de tenis? Apenas se podría adivinar que era sólo el comienzo y que hoy sólo tiene que conectar su Spectrum para revivir aquellos tiempos. Este programa es para dos jugadores, y preste atención a las instrucciones, porque nada se indica por pantalla: la raqueta azul se maneja con la "a" y la "z" y la raqueta amarilla mediante el "1" y "symbol shift".

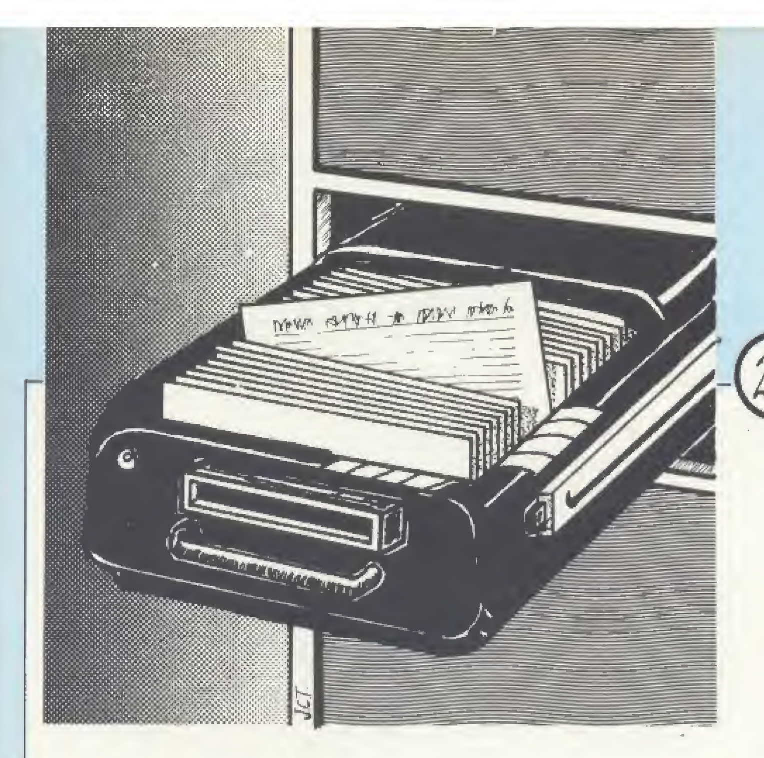
Autor: Jordi Masana *16K*

```
ABCDEFGHIJKLMNOPQRSTU

●BCDEFGHIJKLMNOPQRSTU

  30 GO SUB 1000
  40 BORDER 1: PAPER 4: CLS
  50 LET a=5: LET b=2
  55 LET c=5: LET d=28
  60 LET v=1: LET h=1
  65 LET p1=-1: LET p2=0
  70 FOR i=0 TO 31
  80 PRINT INK 2;AT 1,i;"■";AT 2
1,i;"■";AT 0,i; INK 1;"■": NEXT
i
  90 PRINT AT 21,1; INK 6; PAPER
 2; BRIGHT 1;"TENIS";AT 21,16;"©
 J.MASANA 1984"
 100 LET s=INT (RND*14)+3: LET p
1=p1+1: LET f=4: GO TO 150
 110 LET p2=p2+1: LET s=INT (RND
*14)+3
 120 LET f=26
 150 PRINT INK 2; PAPER 6; BRIGH
T 1;AT 1,2;a$;" ";p1;AT 1,20;y$;
" ";p2
 160 IF p2=11 THEN PRINT AT 8,11
; INK 6; BRIGHT 1; FLASH 1; PAPE
R 2;"GANADOR ";y$: GO TO 2000
 170 IF p1=11 THEN PRINT AT 8,11
; INK 6; FLASH 1; BRIGHT 1; PAPE
R 2;"GANADOR ";a$: GO TO 2000
 200 IF c>3 AND (IN 49150=253 OR
 IN 49150=189) THEN LET c=c-2: P
RINT AT c+2,d;" ";AT c+3,d;" "
 210 IF c<19 AND (IN 32766=253 O
R IN 32766=189) THEN LET c=c+2:
PRINT AT c-2,d;" ";AT c-3;d;" "
 220 IF a>3 AND IN 65022=254 THE
N LET a=a-2: PRINT AT a+3,b;" ";
AT a+2,b;" "
 230 IF a<19 AND IN 65278=253 TH
EN LET a=a+2: PRINT AT a-2,b;" "
;AT a-3,b;" "
 240 PRINT AT c,d; INK 6; BRIGHT
 1;"■";AT c+1,d;"■";AT c-1,d;"■"
 250 PRINT AT a,b; BRIGHT 1; INK
 1;"■";AT a+1,b;"■";AT a-1,b;"■"
 260 LET s=s+v: LET f=f+h
 265 PRINT AT s-v,f-h;" ";AT s,f
;"●"
 270 IF s=20 THEN LET v=-v: BEEP
 .1,25
 280 IF s=2 THEN LET v=-v: BEEP
.1,25
 290 IF (a=s AND b=f) OR (c=s AN
D d=f) THEN LET h=-h: BEEP .1,15
 300 IF (a+1=s AND b=f) OR (c+1=
s AND d=f) THEN LET h=-h-1: BEEP
 .1,15
 310 IF (a-1=s AND b=f) OR (c-1=
s AND d=f) THEN LET h=-h-1: BEEP
 .1,15
 320 IF f>28 THEN BEEP .1,1: GO
TO 90
 330 IF f<2 THEN BEEP .1,1: GO T
O 110
 340 IF h=-3 THEN LET h=h+1
 350 IF h=0 THEN LET h=h-1
 370 IF h=3 THEN LET h=h-1
 400 GO TO 190
1000 FOR k=0 TO 7
1010 READ x: POKE USR "a"+k,x: N
EXT k
1020 DATA 60,126,255,255,255,255
,126,60
1100 INPUT "nombre del jugador a
zul? ";a$: BEEP 1,12
1200 INPUT "nombre del jugador a
marillo? ";y$: BEEP 1,12
1500 RETURN
2000 PRINT AT 14,2;"OTRO PARTIDI
TO? (s/n)"
2010 IF INKEY$="s" THEN GO SUB 1
100: GO TO 40
2020 IF INKEY$="n" THEN STOP
2030 GO TO 2000
```





Programas

24



Arturo: «Soy un fanático de los ordenadores y adoro la programación».

Archivo

Puede que usted sea de los que utiliza las revistas para leer y tirar. Si por el contrario le gusta coleccionarlas, lo más adecuado es mantener un pequeño fichero con los temas que le interesan. El joven autor de este programa vio en el Spectrum una solución a la ordenación de las revistas de ciencias, fácilmente ampliable a cualquier otra publicación.

El programa sirve para archivar los artículos referentes a un tema, y de cada artículo se introduce el nombre de la revista en la que aparece y la fecha de la misma.

Para crear un tema con sus respectivos datos, se usará la opción A. Al hacerlo se le preguntará por el nombre del tema y se abrirá un canal para el *microdrive* en el *buffer* 4 con el nombre del artículo, el de la revista y la fecha de ésta.

La opción B es el listado del archivo por pantalla, mientras que la opción C utiliza la impresora para mostrar los datos conservando el formato de presentación en pantalla.

Las opciones D y E permiten la necesaria actualización de la información.

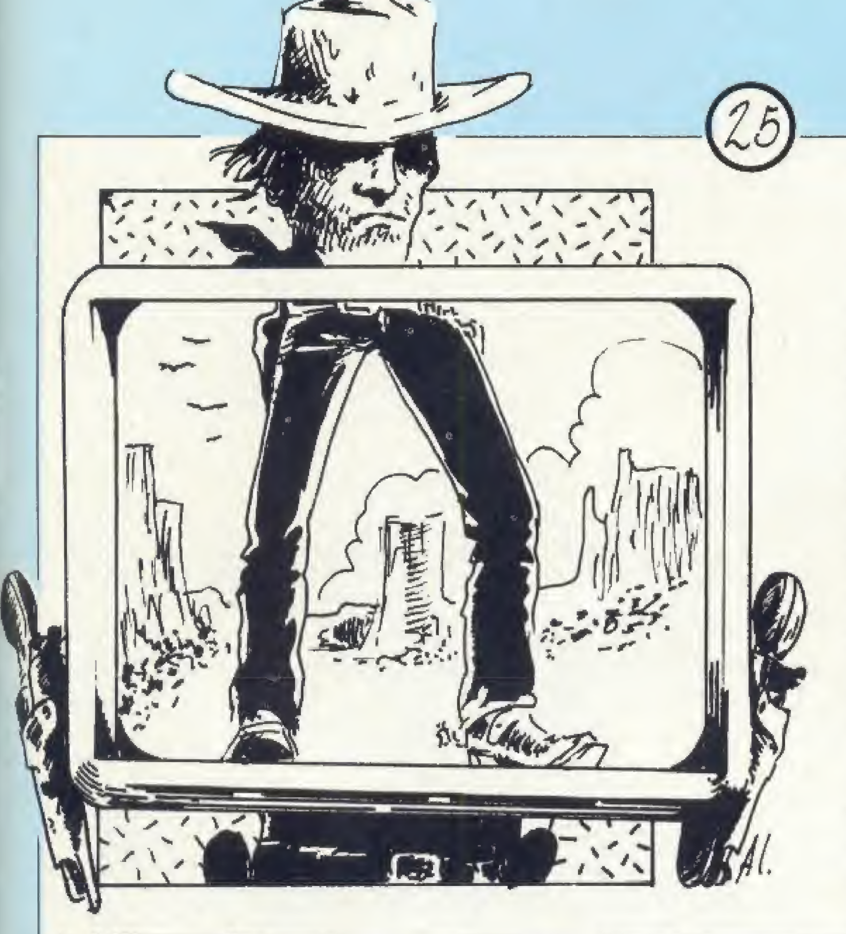
Autor: Arturo García *16K*

```
   1 REM Arturo Garcia Ares. SEP
T-1984
  10 GO SUB 2000
  15 PRINT AT 1,1;"Este es un pr
ograma para      ##almacenar en
MICRODRIVE :";AT 4,1;"1)-las rev
istas,";AT 5,1;"3)-la fecha";AT
6,1;"4)-y el nombre del articulo
";AT 9,1;"de los temas mas inter
esantes ##de sus revistas"
  17 PRINT AT 16,1;"Pulse una te
cla": PAUSE 0
  30 CLS : GO SUB 2000
  32 PRINT AT 5,4; FLASH 1;"A";A
T 8,4;"B";AT 11,4;"C";AT 14,4;"D
";AT 17,4;"E"
  35 PRINT AT 5,6;"CREAR TEMA";A
T 8,6;"LISTAR TEMA";AT 11,6;"IMP
RIMIR TEMA";AT 14,6;"ACTUALIZAR
TEMA";AT 17,6;"BORRAR TEMA"
  40 PAUSE 0: LET l$=INKEY$: POK
E 23658,8
  50 IF l$="A" THEN GO TO 100
  60 IF l$="B" THEN GO TO 400
  70 IF l$="C" THEN GO TO 600
  80 IF l$="D" THEN GO TO 800
  85 IF l$="E" THEN GO TO 1000
  90 GO TO 40
 100 CLS
 120 POKE 23658,8
 130 INPUT "NOMBRE DEL NUEVO TEM
A? ";T$: IF LEN T$>20 THEN PRINT
 AT 10,0; FLASH 1;"Nombre muy la
rgo";AT 21,0;"Pulse una tecla":
PAUSE 0: CLS : GO TO 130
 132 LET d$=t$
 137 IF LEN t$>10 THEN LET t$=t$
( TO 10)
 140 PRINT "AHORA VAS A INTRODUC
IR LOS DATOS SEGUN TE LOS PREGUN
TE. CUANDO  ACABES PON UN PUNTO
COMO NOMBRE DEL ARTICULO.": PRIN
T AT 21,0;"Pulse una tecla": PAU
SE 0
 145 CLS
 150 OPEN #4;"m";1;t$
 160 INPUT "SECCION O NOMBRE DEL
 ARTICULO ?  ";P$: IF LEN P$>30
THEN GO TO 160
 165 IF CODE P$=46 THEN GO TO 30
0
 170 INPUT " REVISTA EN LA QUE
APARECE EL  TEMA : ";H$
 180 INPUT "FECHA : ";F$
 190 CLS : PRINT "TEMA: ";d$''"N
OMBRE DEL ARTICULO: ";P$'"REVIST
A: ";H$'"FECHA : ";F$
 195 INPUT "TODO CORRECTO (S/N)?
 ";C$: IF C$="N" THEN CLS : GO T
O 160
 197 IF C$<>"S" THEN GO TO 195
 200 PRINT #4;P$'H$'F$
```

```
 250 GO TO 160
 300 PRINT #4;".": CLOSE #4
 350 GO TO 30
 400 POKE 23658,8
 410 CLOSE #4: CLS : INPUT "NOMB
RE DEL TEMA A LEER? ";I$: IF LEN
 I$>20 THEN PRINT "LONGITUD no c
orrecta": PAUSE 200: GO TO 410
 415 LET O$=I$
 420 IF LEN I$>10 THEN LET I$=I$
( TO 10)
 430 OPEN #4;"m";1;I$
 435 PRINT "CADA VEZ QUE PRESION
E UNA TECLA SE LLENARA LA PANTAL
LA DE DATOS  (ARTICULO, REVISTA
Y FECHA)": PRINT AT 21,0;"Pulse
una tecla": PAUSE 0: CLS
 436 LET b=4: LET E$="**********
**********************"
 438 GO SUB 4000
 439 FOR n=b TO 24 STEP 2
 440 INPUT #4;P$: IF CODE P$=46
THEN PAUSE 0: GO TO 550
 450 INPUT #4;H$;F$
 480 PRINT ".-";P$
 485 IF SCREEN$ (21,0)<>" " THEN
 GO TO 560
 490 PRINT "* ";H$;" * ";
 500 PRINT "( ";F$;" )"
 510 PRINT AT n,0; OVER 1;"______
_________________________"
 520 NEXT n
 550 CLOSE #4: GO TO 30
 560 PAUSE 0: CLS : GO SUB 3000
 570 GO TO 439
 620 INPUT "NOMBRE DEL TEMA A IM
PRIMIR ? ";I$: IF LEN I$>20 THEN
 PRINT "Lomgitud no correcta": P
AUSE 200: GO TO 410
 625 LET O$=I$
 630 IF LEN I$>10 THEN LET I$=I$
( TO 10)
 640 OPEN #4;"m";1;I$
 650 LPRINT "TEMA : *";O$;"*"
 660 LPRINT
 670 LPRINT "********************
**************"
 680 INPUT #4;P$: IF CODE P$=46
THEN CLOSE #4: GO TO 30
 685 INPUT #4;H$;F$
 690 LPRINT ".-";P$
 700 LPRINT "* ";H$;" * ";
 710 LPRINT "( ";F$;" )"
 720 LPRINT "-------------------
--------------"
 730 GO TO 680
 800 POKE 23658,8: CLS
 820 INPUT "NOMBRE DEL TEMA A AC
TUALIZAR ? ";I$: IF LEN I$>20 T
HEN PRINT "LONGITUD no correcta"
: PAUSE 200: GO TO 810
 825 LET O$=I$
 830 IF LEN I$>10 THEN LET I$=I$
( TO 10)
 840 OPEN #4;"m";1;I$
 850 OPEN #5;"m";1;"ampliacion"
 860 INPUT #4;P$: IF CODE P$=46
THEN GO TO 895
 870 INPUT #4;H$;F$
 880 PRINT #5;P$'H$'F$
 890 GO TO 860
 895 CLOSE #4
 899 ERASE "m";1;I$: PRINT "AHOR
A INTRODUCIRA LOS DATOS     SEGU
N SE LOS PREGUNTE.";AT 21,0;"Pul
se una tecla": PAUSE 0
 900 CLS : INPUT "NOMBRE DEL ART
ICULO ? ";P$: IF CODE P$=46 THEN
 GO TO 970
 910 INPUT "REVISTA ? ";H$
 920 INPUT "FECHA ? ";F$
 940 CLS : PRINT "TEMA : *";O$;"
*"''"ARTICULO : ";P$'"REVISTA :
";H$'"FECHA : ";F$: INPUT "TODO
CORRECTO (S/N) ? ";T$: IF T$="N"
 THEN GO TO 900
 950 IF T$<>"S" THEN GO TO 940
 960 PRINT #5;P$'H$'F$
 965 GO TO 900
 970 PRINT #5;"."
 971 CLOSE #5
 972 OPEN #5;"m";1;"ampliacion"
 973 OPEN #4;"m";1;I$
 974 INPUT #5;P$: IF CODE P$=46
THEN GO SUB 976: GO TO 30
 975 GO TO 981
 976 PRINT #4;"."
 977 CLOSE #5
 978 CLOSE #4
 979 ERASE "m";1;"ampliacion"
 980 RETURN
 981 INPUT #5;H$;F$
 985 PRINT #4;P$'H$'F$
 990 GO TO 974
1000 CLS : INPUT "SABE EL NOMBRE
 DEL TEMA A BORRAR ? (S/N) ";S$;
 IF S$="N" THEN CAT 1: PRINT ''
"CUENTE SOLO LOS NOMBRES DE TEMA
S": PRINT '''"Pulse una tecla":
PAUSE 0: GO TO 1005
1003 IF S$<>"S" THEN GO TO 1000
1005 CLS : INPUT "NOMBRE DEL TEM
A A BORRAR ? ";J$: IF LEN J$>20
THEN GO TO 1000
1010 PRINT "ESTA SEGURO DE QUE Q
UIERE BORRAR ";J$;" ? (S/N)"
1020 INPUT L$: IF L$="S" THEN GO
 TO 1050
1030 IF l$<>"N" THEN GO TO 1010
1040 GO TO 30
1050 IF LEN J$>10 THEN LET J$=J$
(1 TO 10)
1060 ERASE "m";1;J$
1070 GO TO 30
2000 LET s$="###################
#############": PRINT AT 0,0;s$;
AT 21,0;s$
2010 FOR f=1 TO 20: PRINT AT f,0
;"#";AT f,31;"#": NEXT f
2020 RETURN
3000 LET b=6
3010 GO SUB 4000
3020 PRINT AT 3,0;".-";P$
3030 PRINT AT 4,0;"* ";H$;" * ";
3040 PRINT "( ";F$;" )"
3050 PRINT AT 4,0; OVER 1;"_____
_________________________"
3060 RETURN
4000 PRINT "TEMA : *";O$;"*"
4010 PRINT AT 2,0;E$
4020 RETURN
```

Programas ㉕

Far-West



Si es usted un intrépido aventurero de los que desconoce el significado de la palabra "miedo", acaba de encontrar un programa hecho a su medida. El objetivo es derribar al contrario desenfundando antes. ¿Que cuál es el motivo? Eso no importa demasiado: cualquier motivo es bueno. ¿Cuál es su contrincante? Eso sí importa y mucho, porque si elige al temido Spectrum ya puede ir cavando su sepultura. Le aconsejamos empiece con sus amigos, pero no le daremos la más mínima indicación de cómo usar el programa: él solito se lo cuenta todo. No falta la música del legendario oeste (no se pierda la sintonía de la rutina que comienza en la línea 27). ¡Que la suerte le acompañe, forastero!

Debido a la complejidad de los caracteres gráficos definidos, el listado 2 es una copia de las líneas 150 a 480 previo a la definición, para mayor sencillez en el segmento de los caracteres gráficos empleados.

Autor: Antonio Garrido *16K*

LISTADO 1

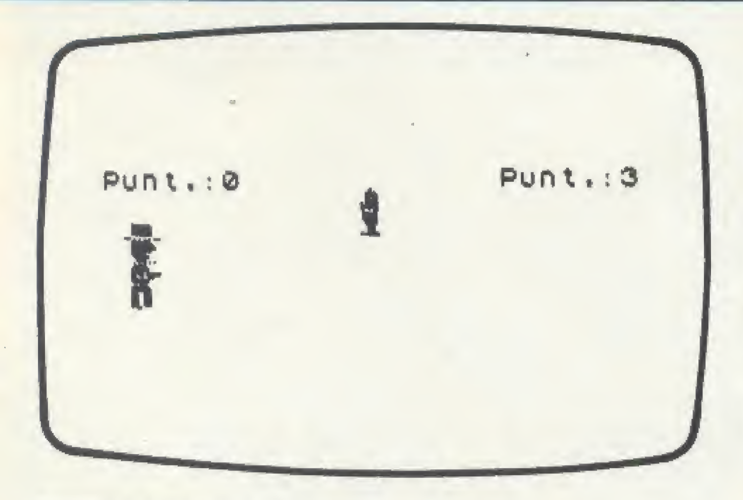
```
   1 GO SUB 600
   3 BORDER 0: PAPER 0: INK 7: C
LS
   5 LET q=0: LET sp=-1
   6 CLS : PRINT TAB 10;"Far-wes
t";AT 0,10; OVER 1;"_________"
  10 LET p=0: PRINT ''''" Bienven
ido al Far-west donde launica le
y la impone el mas rapi-do.""."
 PRINT AT 21,0;"Instrucciones (S
/N)?": GO SUB 27: LET a$=p$: IF
a$<>"n" AND a$<>"N" AND a$<>"s"
AND a$<>"S" THEN PRINT AT 1,0;:
GO TO 10
  11 LET p=0: PRINT AT 21,0;"Ver
sion de dos jugadores?": GO SUB
27: LET k$=p$: IF k$<>"s" AND k$
<>"S" AND k$<>"n" AND k$<>"N" TH
EN GO TO 11
  12 IF k$="n" OR k$="N" THEN LE
T k$=""
  13 IF a$="n" OR a$="N" THEN GO
 TO 60
  14 PRINT AT 9,0;" Pasando por
Creeksville, te hasvisto envuelt
o en una pelea al  intentar salv
ar a la dulce        Sweet Lucy Lo
ve (La chica del   salon). Y lo
que es peor, al    amanecer habr
as de vertelas con el pistolero
mas temido de todoslos tiempos."
  15 PRINT AT 21,0;"Presione 'EN
TER' si esta listo.": LET p=0: G
O SUB 27
  16 IF k$="s" OR k$="S" THEN CL
S : PRINT TAB 10;"Far-west";AT 0
,10; OVER 1;"_________": PRINT '
'"Jugador uno:-": GO TO 36
  17 IF k$="" THEN GO TO 20
  18 GO TO 15
  20 CLS : PRINT TAB 10;"Far-wes
t";AT 0,10; OVER 1;"_________"
  21 PRINT ''''" Su oponente es e
l temible        SPECTRUM."
  22 PRINT " Es extremadamente p
reciso no   errando ningun dispa
ro."
  23 PRINT '"Tu unica esperanza
es escondertedetras de los cactu
s y las ca-  rretas para intenta
r sorprender-le."
  24 LET p=0: PRINT AT 21,0;"Pre
sione 'ENTER' si esta listo.": G
O SUB 27
  25 CLS : PRINT TAB 10;"Far-wes
t";AT 0,10; OVER 1;"_________"
  26 GO TO 36
  27 LET p$="": LET sp=sp+1: REA
D h,i: IF i=-1.7 THEN GO TO 32
  28 BEEP h,i: LET p=p+1: IF p>3
 THEN LET p$=INKEY$: IF p$<>"" T
HEN RETURN
  29 DATA .07,15.5,.05,20.6,.05,
15.5,.05,20.6,1,15.5,.5,11.5,.5,
13.5,1.2,8.5
  30 DATA .07,15.5,.05,20.6,.05,
15.5,.05,20.6,1,15.5,.5,11.5,.5,
13.5,1.2,18.5
  31 DATA .07,15.5,.05,20.6,.05,
15.5,.05,20.6,1,15.5,.5,11.5,.2,
10.5,.2,8.5,1.2,6.5,.07,15.5,.05
,20.6,.05,15.5,.05,20.6,1,15.5,.
5,13.5,1.5,8.5
  32 IF i=-1.7 THEN RESTORE 29:
FOR p=1 TO 200: LET p$=INKEY$: I
F p$="" THEN NEXT p
  33 IF p$<>"" THEN RETURN
```

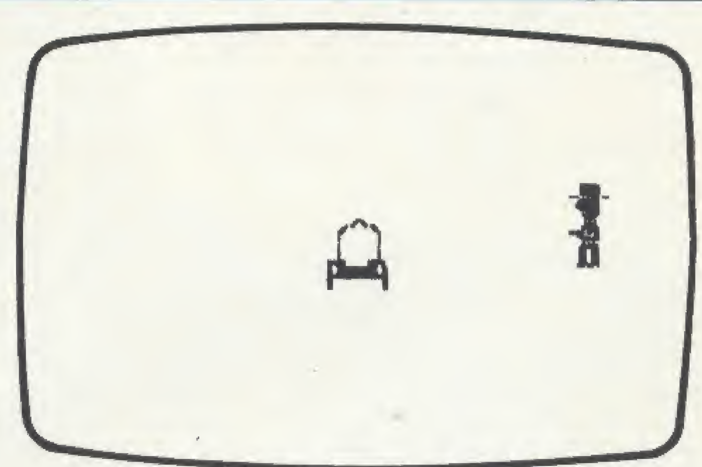
Programas

(25)

```
  34 GO TO 27
  36 PRINT '''"Tu cowboy es el d
e la derecha.",''"Para ir hacia
arriba=teclas  6 ala 0"
  37 PRINT '"Para ir hacia abajo
=teclas de laY a la P"
  38 PRINT '"Para disparar=tecla
s de la B a  SPACE."
  40 LET p=0: PRINT AT 21,0;"Pre
sione 'ENTER' si esta listo.": G
O SUB 27
  41 IF K$="" THEN GO TO 60
  42 CLS : PRINT TAB 10;"Far-wes
t";AT 0,10; OVER 1;"________"
  43 PRINT ''"Jugador dos:-"
  44 PRINT '''" Tu cowboy es el
de la izquier-  da.",''"Para ir
hacia arriba=  teclas de la 1 a
la 5"
  45 PRINT '"Para ir hacia abajo
=teclas de laQ a la T."
  46 PRINT '"Para disparar=tecla
s de CAPS    SHIFT a la V"
  47 LET p=0: PRINT AT 21,0;"Pre
sione 'ENTER' si esta listo.": G
O SUB 27
  60 CLS : PRINT TAB 10;"Far-wes
t";AT 0,10; OVER 1;"________"
  65 LET p=0: PRINT AT 9,0;"Pres
ione Y para comenzar."''"Presion
e N para recomenzar."''"Presione
 S para parar": GO SUB 27
  66 GO TO (p$="s" OR p$="S")*2+
(p$="y" OR p$="Y")*4+(p$="n" OR
p$="N")*(-60)+65
  67 STOP
  69 IF K$="" THEN CLS : PRINT T
AB 10;"Far-west";AT 0,10; OVER 1
;"________";AT 9,2;"BUENA SUERTE
...LA NECESITARAS!": FOR A=1 TO
200: NEXT A
  70 PAPER 1: INK 7: BORDER 0: C
LS
  80 LET s1=0: LET s2=s1
  90 FOR g=1 TO 5
 100 LET T=18: LET R=-2: LET G=0
 110 CLS : LET a=INT (RND*6)+2:
LET b=4: LET c=INT (RND*6)+10: L
ET D=27: LET a1=a: LET c1=c
 120 PRINT AT 0,3;"Punt.:";AT 0,
23;"Punt.:"
 140 GO SUB 300: GO SUB 330
 150 IF Q=2 OR Q=4 THEN PRINT AT
 18,15;"┣━┫";AT 17,15;"┴ ┴";AT 1
6,15;"⌒"
 160 FOR J=1 TO 20 STEP 3: IF Q=
2 OR Q=4 THEN PRINT AT 1,16; INK
 6;"♟";AT 2,16; INK 6;"♜";AT 19,
16; INK 6;"♟";AT 20,16; INK 6;"♜
": GO TO 180
 170 IF RND>.4 THEN PRINT AT J,1
6;; INK 6;"♟";AT J+1,16; INK 6;"
♜"
 180 NEXT J
 185 REM Lectura teclado.
 190 PRINT AT 0,9;s1;AT 0,29;s2:
 LET g=(IN 65278<>255): LET a1=a
-(2*(IN 63486<>255))+(2*(IN 6451
0<>255)): LET C1=C-(2*(IN 61438<
>255))+(2*(IN 57342<>255)): LET
f=(IN 32766<>255)
 200 IF Q=2 OR Q=4 THEN PRINT AT
 T,15;"   ";AT T-1,15;"   ";AT T
-2,15;"   ": LET T=T+R: PRINT AT
 T,15;"┣━┫";AT T-1,15;"┴ ┴";AT T
-2,15;"⌒": LET R=(4*(R+T<6)-2)
+(4+(R=2))-(4*(R+T>18)): PAUSE 1
3
 210 IF c1<19 AND c1>0 AND c1<>c
 THEN GO SUB 320
 220 LET c1=c
 230 GO TO f*110+240
 240 IF k$="" THEN GO SUB 390
 250 IF a1<19 AND a1>0 AND a1<>a
 THEN GO SUB 290
 260 LET a1=a
 270 GO TO g*90+280
 280 GO TO 190
 290 GO SUB 300
 300 PRINT OVER 1;AT A,B; INK 5;
"▄";AT A+1,B;"♥";AT A+2,B; INK
5;"♣";AT A+3,B; INK 5;"▀"
 310 LET a=a1: RETURN
 320 GO SUB 330
 330 PRINT OVER 1;AT C,D;"▄";AT
 C+1,D;"♥";AT C+2,D;"♣";AT C+3
,D;"▀"
 340 LET c=c1: RETURN
 345 REM Disparos.
 350 BEEP .0059,25: PLOT 216,155
-(c*8): LET e=POINT (133,155-c*8
): LET H=POINT (140,155-C*8): DR
AW -84,0: DRAW OVER 1; INK 6;83,
0: PLOT OVER 1; INK 6;132,155-(c
*8): IF e=0 AND H=0 THEN DRAW -9
5,0: DRAW OVER 1; INK 6;94,0: PL
OT OVER 1;37,155-(c*8): LET a$=S
CREEN$ (c+2,4): GO TO ((a$<>" ")
*180)+240
 360 GO TO 240
 370 BEEP .0059,30: PLOT INK 5;4
7,155-(a*8): LET e=POINT (131,15
5-a*8): LET H=POINT (124,155-A*8
): DRAW 84,0: DRAW OVER 1; INK 6
;-83,0: PLOT OVER 1; INK 6;131,1
55-(a*8): IF e=0 AND H=0 THEN DR
AW 88,0: DRAW OVER 1; INK 6;-87,
0: PLOT OVER 1;219,155-(a*8): LE
T a$=SCREEN$ (a+2,27): GO TO ((a
$<>" ")*180)+280
 380 GO TO 280
 390 IF A>C1+2 OR A<C1 THEN IF R
ND>.5 THEN LET A1=A-(2*(A>C1+3))
+(2*(A<C1))
 400 IF a1+2>=c1 AND a1+2<=c1+3
THEN LET g=1
 410 RETURN
 420 RESTORE 550: LET s2=s2+1: P
RINT AT a,b;" ";AT a+2,b;" ";A
T a+1,b-1;"O.K.";AT a+3,b-1;"███
███": FOR W=1 TO 40: NEXT W
 430 IF A<13 THEN PRINT AT a+1,b
-1;"    ";AT a+3,b-1;"    ":
 LET A=13: PRINT AT A,3;"R.I.P."
;AT A+2,3;"██████"
 440 FOR Z=A+3 TO A-9 STEP -1: G
O SUB 500: PRINT AT Z,3;"      "
;AT Z-2,3;"      ";AT Z-3,3;"R.I
.P.";AT Z-1,3;"██████": NEXT Z
 450 GO TO 560
 460 RESTORE 520: LET s1=s1+1: P
RINT AT C,D;" ";AT c+1,d;" ";A
T c+3,d;" ";AT C+2,D;" ";AT C+
1,D-3;"O.K.";AT C+3,d-3;"██████"
: FOR W=1 TO 40: NEXT W
 470 IF C<11 THEN PRINT AT C+1,D
```


Programas/Juegos/Montajes/Código Màquina
AÑO 2 - No. 13 / Diciembre 1984 250 Ptas.
ZX
REVISTA PARA LOS USUARIOS DE ORDENADORES SINCLAIR
YA ESTA A LA VENTA
GUÍA DE SOFTWARE PARA EL SPECTRUM
REGALAMOS ESTA MOTO
vespino
Visitamos Sinclair Research





LISTADO 2

```
 150 IF Q=2 OR Q=4 THEN PRINT AT
18,15;"D▀E";AT 17,15;"F F";AT 1
6,15;"ABC"
 160 FOR J=1 TO 20 STEP 3: IF Q=
2 OR Q=4 THEN PRINT AT 1,16; INK
 6;"H";AT 2,16; INK 6;"I";AT 19,
16; INK 6;"H";AT 20,16; INK 6;"I
": GO TO 180
 170 IF RND>.4 THEN PRINT AT J,1
6;; INK 6;"H";AT J+1,16; INK 6;"
I"
 180 NEXT J
 185 REM Lectura teclado.
 190 PRINT AT 0,9;s1;AT 0,29;s2:
 LET g=(IN 65278<>255): LET a1=a
-(2*(IN 63486<>255))+(2*(IN 6451
0<>255)): LET C1=C-(2*(IN 61438<
>255))+(2*(IN 57342<>255)): LET
f=(IN 32766<>255)
 200 IF Q=2 OR Q=4 THEN PRINT AT
 T,15;"   ";AT T-1,15;"   ";AT T
-2,15;"   ": LET T=T+R: PRINT AT
 T,15;"D▀E";AT T-1,15;"F F";AT T
-2,15;"ABC": LET R=(4*(R+T<6)-2)
+(4+(R=2))-(4*(R+T>18)): PAUSE 1
3
 210 IF c1<19 AND c1>0 AND c1<>c
 THEN GO SUB 320
 220 LET c1=c
 230 GO TO f*110+240
 240 IF k$="" THEN GO SUB 390
 250 IF a1<19 AND a1>0 AND a1<>a
 THEN GO SUB 290
 260 LET a1=a
 270 GO TO g*90+260
 280 GO TO 190
 290 GO SUB 300
 300 PRINT OVER 1;AT A,B; INK 5;
"▄ ";AT A+1,B;"ON";AT A+2,B; INK
 5;"RS";AT A+3,B; INK 5;"TU"
 310 LET a=a1: RETURN
 320 GO SUB 330
 330 PRINT OVER 1;AT C,D;" ▄";AT
 C+1,D;"QP";AT C+2,D;"KJ";AT C+3
,D;"ML"
 340 LET c=c1: RETURN
 345 REM Disparos.
 350 BEEP .0059,25: PLOT 216,155
-(c*8): LET e=POINT (133,155-c*8
): LET H=POINT (140,155-C*8): DR
AW -84,0: DRAW OVER 1; INK 6;83,
0: PLOT OVER 1; INK 6;132,155-(c
*8): IF e=0 AND H=0 THEN DRAW -9
5,0: DRAW OVER 1; INK 6;94,0: PL
OT OVER 1;37,155-(c*8): LET a$=S
CREEN$ (c+2,4): GO TO ((a$<>" ")
*180)+240
 360 GO TO 240
 370 BEEP .0059,30: PLOT INK 5;4
7,155-(a*8): LET e=POINT (131,15
5-a*8): LET H=POINT (124,155-A*8
): DRAW 84,0: DRAW OVER 1; INK 6
;-83,0: PLOT OVER 1; INK 6;131,1
55-(a*8): IF e=0 AND H=0 THEN DR
AW 88,0: DRAW OVER 1; INK 6;-87,
0: PLOT OVER 1;219,155-(a*8): LE
T a$=SCREEN$ (a+2,27): GO TO ((a
$<>" ")*180)+280
 380 GO TO 280
 390 IF A>C1+2 OR A<C1 THEN IF R
ND>.5 THEN LET A1=A-(2*(A>C1+3))
+(2*(A<C1))
 400 IF a1+2>=c1 AND a1+2<=c1+3
THEN LET g=1
 410 RETURN
 420 RESTORE 550: LET s2=s2+1: P
RINT AT a,b;"  ";AT a+2,b;"  ";A
T a+1,b-1;"O.K.";AT a+3,b-1;"███
███": FOR W=1 TO 40: NEXT W
 430 IF A<13 THEN PRINT AT a+1,b
-1;"       ";AT a+3,b-1;"      ":
 LET A=13: PRINT AT A,3;"R.I.P."
;AT A+2,3;"██████"
 440 FOR Z=A+3 TO A-9 STEP -1: G
O SUB 500: PRINT AT Z,3;"      "
;AT Z-2,3;"      ";AT Z-3,3;"R.I
.P.";AT Z-1,3;"██████": NEXT Z
 450 GO TO 560
 460 RESTORE 520: LET s1=s1+1: P
RINT AT C,D;"  ";AT c+1,d;"  ";A
T c+3,d;"  ";AT C+2,D;"  ";AT C+
1,D-3;"O.K.";AT C+3,d-3;"██████"
: FOR W=1 TO 40: NEXT W
 470 IF C<11 THEN PRINT AT C+1,D
-3;"       ";AT C+3,D-3;"      ":
 LET C=11: PRINT AT C,24;"R.I.P.
";AT C+2,24;"██████"
 480 FOR Z=C+3 TO C-7 STEP -1: G
O SUB 500: PRINT AT Z,24;"
";AT Z-2,24;"      ";AT Z-3,24;"
R.I.P.";AT Z-1,24;"██████": NEXT
 Z
```

Programas

25

```
 490 GO TO 560
 495 REM Musica de R.I.P.
 500 READ X,Y: BEEP X,Y
 510 RETURN
 520 DATA 1,-1.7,1,-1.7,.2,-1.7,
1,-1.7,.5,1.4,.5,.35,.5,.35,.6,-
1.7,1,-1.7,.2,-1.7,1,-1.7
 550 DATA .2,-2,.2,1,.5,1,.2,-1,
.2,3,.5,3,.2,5,.2,5,.2,3,.2,5,.2
,6,.2,8,.6,10
 560 NEXT q
 565 PRINT AT 0,9;51;AT 0,29;52
 566 FOR A=1 TO 25: NEXT A: REST
ORE 556: FOR A=1 TO 11: READ B,C
: BEEP B,C: NEXT A
 567 DATA .2,1,.09,-60,.1,-4,.1,
-4,.1,-4,.4,-3,.58,-4,.1,-60,.3,
0,.09,-60,.2,1
 570 POKE 23659,1: PRINT AT 22,0
; PAPER 0;"Probamos de nuevo? ":
 POKE 23659,2: LET a$=INKEY$: IF
 a$="" THEN GO TO 570
 580 IF A$="N" OR A$="n" THEN ST
OP
 590 IF a$="s" OR a$="S" THEN RU
N
 595 GO TO 570
 599 REM Definicion caracteres g
raficos.
 600 RESTORE 610: FOR a=0 TO 167
: READ b: POKE USR "a"+a,b: NEXT
 a: RESTORE : RETURN
 610 DATA 0,0,1,6,12,8,8,8,28,54
,99,0,0,0,0,0,0,96,48,8,8,8,8
 620 DATA 207,207,207,255,192,19
2,192,192,243,243,243,255,3,3,3,
3,8,8,8,8,8,8,254,214
 630 DATA 126,126,110,62,28,28,2
8,255,8,28,62,54,62,186,254,214,
126,126,110,62,28,28,28,255
 640 DATA 160,224,240,240,176,48
,240,64,2,3,7,5,253,24,7,5,240,2
40,112,112,112,112,112,208
 650 DATA 7,15,14,14,14,14,14,29
,87,240,216,252,240,128,224,192,
117,15,15,15,15,15,7,3
 660 DATA 87,240,240,240,240,240
,224,192,245,15,27,63,15,1,7,3,5
,7,15,15,13,12,15,5
 670 DATA 84,192,224,160,191,24,
224,64,15,15,14,14,14,14,14,11,2
24,240,112,112,112,112,112,0
```

Programas (26)

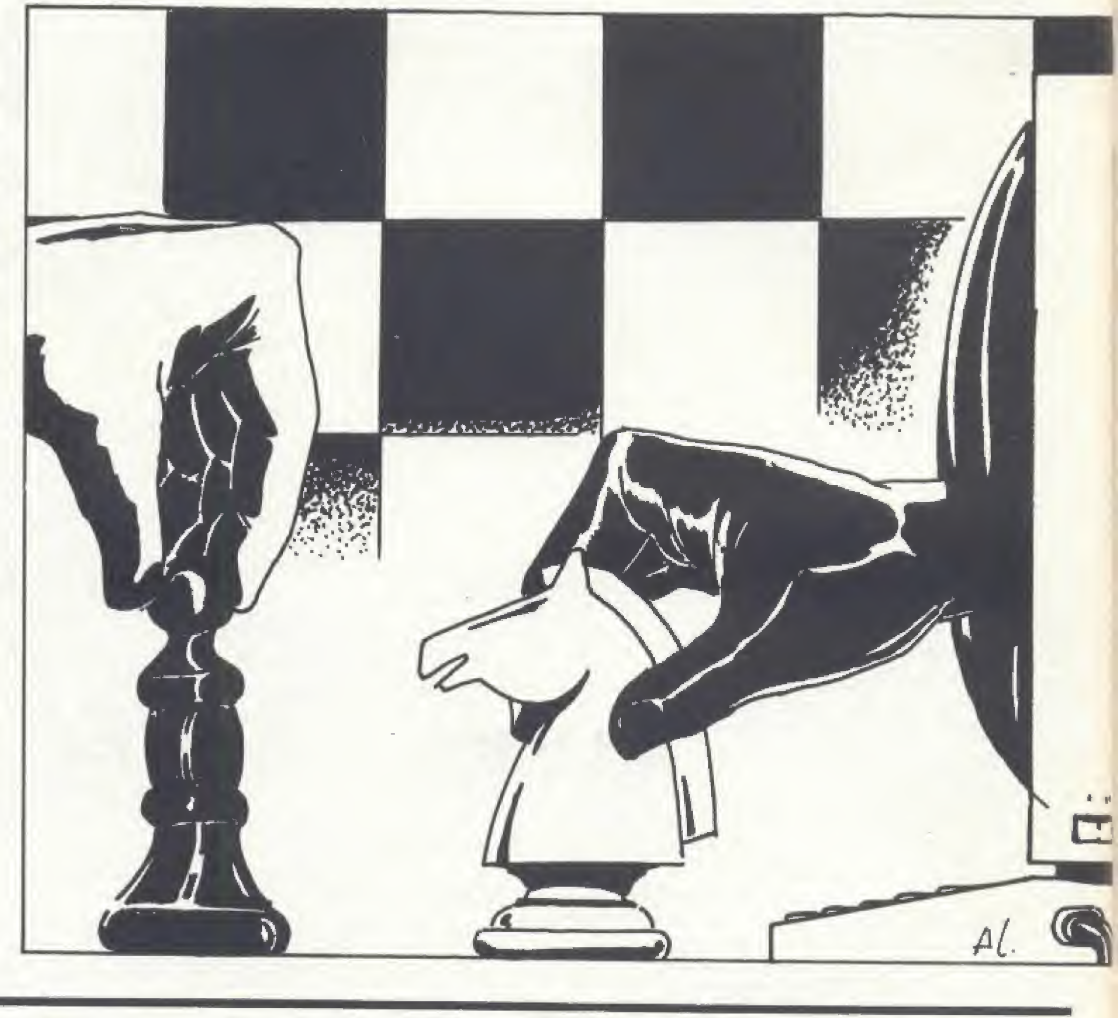
Ajedrez

Con este programa, como ya habrá adivinado, podrá jugar al ajedrez. Pero no se entusiasme demasiado: sólo sustituye al tablero, ya que no podrá jugar contra el ordenador. En cualquier caso, es un bonito programa con una definición gráfica inmejorable. Para mover las fichas ha de indicar la ficha que desea mover, las coordenadas de la posición en que se encuentre y las nuevas coordenadas a donde se desea llevarla. Después presione "e" si se equivocó o "m" si es correcto. Empiezan, como todos saben, las blancas.

```
ABCDEFGHIJKLMNOPQRSTU
♝B♞♛EFGHIJKLMNO♟♚♛S♜U
```


Autor:
Juan Hernández Navarro

48K




```
   1 CLEAR : RESTORE
  10 BORDER 1: PAPER 4: INK 1: C
LS
  20 POKE 23609,90
  30 LET a=20: LET b=35
  40 LET c=16: LET d=28
  50 LET w=24: LET x=60
  60 LET y=90: LET z=126
  70 GO SUB 1500
 100 REM Tablero
 110 PLOT a,d
 120 DRAW 0,128: DRAW 128,0
 130 DRAW 0,-128: DRAW -128,0
 140 PLOT a-1,d-1
 150 DRAW 0,130: DRAW 130,0
 160 DRAW 0,-130: DRAW -130,0
 170 FOR n=a TO b
 180 PLOT n,d: DRAW 0,c
 190 NEXT n
 200 LET d=d+32
 210 IF d=156 THEN LET d=44: GO
TO 240
 220 IF d=172 THEN LET d=28: GO
TO 240
 230 GO TO 170
 240 LET a=a+16: LET b=b+16
 250 IF a=148 THEN GO TO 300
 260 GO TO 170
 300 REM Coordenadas
 310 PRINT INK 0;AT 1,3;"A B C D
E F G H"
 320 FOR e=1 TO 8
 330 PRINT INK 0;AT e*2+1,1;CHR$
(48+e)
 340 NEXT e
 400 REM Posicion inicial
 410 PRINT INK 0;AT 3,3;"♜";AT 3
,5;"♞";AT 3,7;"♝";AT 3,9;"♛"
 420 PRINT INK 0;AT 3,11;"♚";AT
3,13;"♝";AT 3,15;"♞";AT 3,17;"♜"
```

```
 430 PRINT INK 7;AT 17,3;"♜";AT
17,5;"♞";AT 17,7;"♝";AT 17,9;"♛"
 440 PRINT INK 7;AT 17,11;"♚";AT
17,13;"♝";AT 17,15;"♞";AT 17,17
;"♜"
 450 FOR l=3 TO 17 STEP 2
 460 PRINT INK 0;AT 5,l;"♟"
 470 PRINT INK 7;AT 15,l;"♟"
 480 NEXT l
 500 REM Introduccion de datos
 510 LET i=7
 520 IF i=7 THEN PRINT INK i;AT
1,22;"BLANCAS";AT 20,19+INT (12-
LEN b$)/2;b$: BEEP .5,20
 530 IF i=0 THEN PRINT INK i;AT
1,21;" NEGRAS";AT 20,19+INT (12-
LEN c$)/2;c$: BEEP .5,20
 540 PAUSE 20
 550 PRINT INK i;AT 4,22;"Pieza:
?": POKE 22685,160
 560 PAUSE 0: LET z$=INKEY$
 570 IF z$<>"r" AND z$<>"d" AND
z$<>"t" AND z$<>"a" AND z$<>"c"
AND z$<>"p" AND z$<>"e" AND z$<>
"q" THEN GO TO 560
 580 PRINT INK i;AT 4,29;CHR$ (C
ODE z$+47): BEEP .1,20
 590 IF z$="e" THEN GO SUB 900:
GO TO 830
 600 IF z$="q" THEN PRINT FLASH
1; INK i;AT 20,8;"Cambio": GO TO
520
 610 PAUSE 20
 620 PRINT INK i;AT 7,22;"Posici
on";AT 8,22;"inicial:";AT 10,23;
"?": POKE 22871,160
 630 PAUSE 0: LET p$=INKEY$
 640 IF p$<"a" OR p$>"h" THEN GO
TO 630
 650 PRINT INK i;AT 10,23;CHR$ (
CODE p$-32);" ?": BEEP .1,30: PO
KE 22873,160
 660 PAUSE 0: LET q$=INKEY$
 670 IF q$<"1" OR q$>"8" THEN GO
TO 660
 680 PRINT INK i;AT 10,25;q$: BE
EP .1,30
 690 PAUSE 20
 700 PRINT INK i;AT 13,22;"Posic
ion";AT 14,22;"final:";AT 16,23;
"?": POKE 23063,160
 710 PAUSE 0: LET r$=INKEY$
 720 IF r$<"a" OR r$>"h" THEN GO
TO 710
 730 PRINT INK i;AT 16,23;CHR$ (
CODE r$-32);" ?": BEEP .1,30: PO
KE 23065,160
 740 PAUSE 0: LET s$=INKEY$
 750 IF s$<"1" OR s$>"8" THEN GO
TO 740
 760 PRINT INK i;AT 16,25;s$: BE
EP .1,30
 770 PAUSE 20
 780 PRINT INK i;AT 20,6;"( E -
M )"
 790 PAUSE 0: LET j$=INKEY$
 800 IF j$<>"e" AND j$<>"m" THEN
GO TO 790
 810 IF j$="e" THEN POKE 23176,1
60: BEEP .3,30: GO SUB 1200: GO
TO 520
 820 IF j$="m" THEN POKE 23180,1
60: BEEP .3,30: GO SUB 1000
 830 IF i=7 THEN LET i=0: GO TO
520
 840 IF i=0 THEN LET i=7: GO TO
520
 900 REM Enroque
 910 PRINT INK i;AT 7,22;"Enroqu
e";AT 9,20;"corto-largo";AT 11,2
3;"(c/l)"
 920 PAUSE 0: LET e$=INKEY$: BEE
P .1,30
 930 IF e$<>"c" AND e$<>"l" THEN
GO TO 920
 940 IF e$="c" AND i=7 THEN POKE
22904,160: PRINT INK 1;AT 17,11
;"█"; INK 4;AT 17,17;"█"; INK i;
AT 17,15;"♚";AT 17,13;"♜"
 950 IF e$="c" AND i=0 THEN POKE
22904,160: PRINT INK 4;AT 3,11;
"█"; INK 1;AT 3,17;"█"; INK i;AT
3,15;"♚";AT 3,13;"♜"
 960 IF e$="l" AND i=7 THEN POKE
22906,160: PRINT INK 1;AT 17,11
;"█";AT 17,3;"█"; INK i;AT 17,7;
"♚";AT 17,9;"♜"
 970 IF e$="l" AND i=0 THEN POKE
22906,160: PRINT INK 4;AT 3,11;
"█";AT 3,3;"█"; INK i;AT 3,7;"♚"
;AT 3,9;"♜"
 980 GO TO 1200
1000 REM Movimiento
1010 LET z$=CHR$ (CODE z$+47)
1020 LET p=2*CODE p$-191
1030 LET r=2*CODE r$-191
1040 LET q=VAL q$*2+1
1050 LET s=VAL s$*2+1
1060 IF q=3 OR q=5 THEN GO TO 10
90
1070 IF q=7 OR q=9 OR q=11 OR q=
13 THEN GO TO 1110
1080 IF q=15 OR q=17 THEN GO TO
1130
1090 IF PEEK (16384+32*(q+1)+p)=
0 THEN GO TO 1150
1100 IF PEEK (16384+32*(q+1)+p)=
255 THEN GO TO 1160
1110 IF PEEK (18432+32*(q-7)+p)=
0 THEN GO TO 1150
1120 IF PEEK (18432+32*(q-7)+p)=
```

INSTRUCCIONES

Se deberan pulsar:

1. La inicial de la pieza que se desea mover.

R=REY ♚	C=CABALLO ♞
D=DAMA ♛	P=PEON ♟
T=TORRE ♜	E=ENROQUE
A=ALFIL ♝	Q=CAMBIO

2. Las coordenadas de la posicion en la que se encuentra su pieza.

(Presione una tecla)

```
255 THEN GO TO 1160
1130 IF PEEK (20480+32*(q-15)+p)
=0 THEN GO TO 1150
1140 IF PEEK (20480+32*(q-15)+p)
=255 THEN GO TO 1160
1150 PRINT INK 4;AT q,p;"█": GO
TO 1170
1160 PRINT INK 1;AT q,p;"█"
1170 PRINT INK i;AT s,r;z$
1200 REM Nueva jugada
1210 FOR f=0 TO 21
1220 FOR g=20 TO 31
1230 PRINT AT f,g;" "
1240 NEXT g: NEXT f
1250 PRINT AT 20,6;"          "
1260 RETURN
1500 REM Piezas
1510 FOR h=1 TO 6: READ a$
1520 FOR j=0 TO 7: READ k
1530 POKE USR a$+j,k
1540 NEXT j: NEXT h
1550 DATA "r",w,102,153,153,y,z,
66,z
1560 DATA "d",195,36,153,y,y,z,6
6,z
1570 DATA "t",y,y,z,x,x,x,z,z
1580 DATA "a",w,36,y,y,36,w,102,
153
1590 DATA "c",w,46,127,239,78,28
,62,127
1600 DATA "p",w,x,x,w,w,x,x,126
1700 REM Presentacion
1710 PRINT INVERSE 1;AT 1,10;"IN
DICACIONES"
1720 PRINT AT 3,1;"Se deberan pu
lsar:"
1730 PRINT AT 5,1;"1. La inicial
 de la pieza que";TAB 4;"se dese
a mover."
1740 PRINT AT 8,5;"R=REY ♔";TAB
18;"C=CABALLO ♞"
1750 PRINT AT 10,5;"D=DAMA ♕";TA
B 18;"P=PEON ♟"
1760 PRINT AT 12,5;"T=TORRE ♜";T
AB 18;"E=ENROQUE"
1770 PRINT AT 14,5;"A=ALFIL ♝";T
AB 18;"O=CAMBIO"
1780 PRINT AT 16,1;"2. Las coord
enadas de la posi-";TAB 4;"cion
en la que se encuentra";TAB 4;"s
u pieza."
1790 PRINT AT 20,6;"(Presione un
a tecla)"
1800 PAUSE 0: CLS
1810 PRINT AT 1,1;"3. Las coorde
nadas de la posi-";TAB 4;"cion d
onde desea trasladar";TAB 4;"su
pieza."
1820 PRINT AT 5,1;"4. Para estab
lecer las coorde-";TAB 4;"nadas
de una pieza, pulsar";TAB 4;"en
primer lugar la coorde-";TAB 4;"
nada alfabetica (ABCDEFGH),";TAB
 4;"y seguidamente la coordena-"
;TAB 4;"da numerica (12345678)."
1830 PRINT AT 12,1;"5. Presionar
 'e' si ha cometi-";TAB 4;"do al
gun error. Se repetira";TAB 4;"l
a tirada."
1840 PRINT AT 16,1;"6. Presionar
 'm' si su tirada";TAB 4;"es cor
recta. Inmediatamente";TAB 4;"se
 efectuara el movimiento."
1850 PRINT AT 20,6;"(Presione un
a tecla)"
1860 PAUSE 0: CLS
1900 REM Entrada
1910 FOR i=7 TO 0 STEP -7
1920 IF i=0 THEN PRINT INK i;AT
3,3;"Negras "
1930 IF i=7 THEN PRINT INK i;AT
3,3;"Blancas"
1940 PRINT INK i;AT 3,15;"♜ ♞ ♝
♛ ♚ ♝ ♞ ♜"
1950 PRINT INK i;AT 5,15;"♟ ♟ ♟
♟ ♟ ♟ ♟ ♟"
1960 PRINT INK i;AT 16,6;"Introd
uzca su nombre"'';TAB 7;"(maximo
 12 letras)"
1970 IF i=7 THEN INPUT AT 1,10;
LINE b$: IF LEN b$>12 THEN GO TO
 1970
1980 IF i=0 THEN INPUT AT 1,10;
LINE c$: IF LEN c$>12 THEN GO TO
 1980
1990 NEXT i
2000 CLS : RETURN
9999 SAVE "AJEDREZ" LINE 1
```

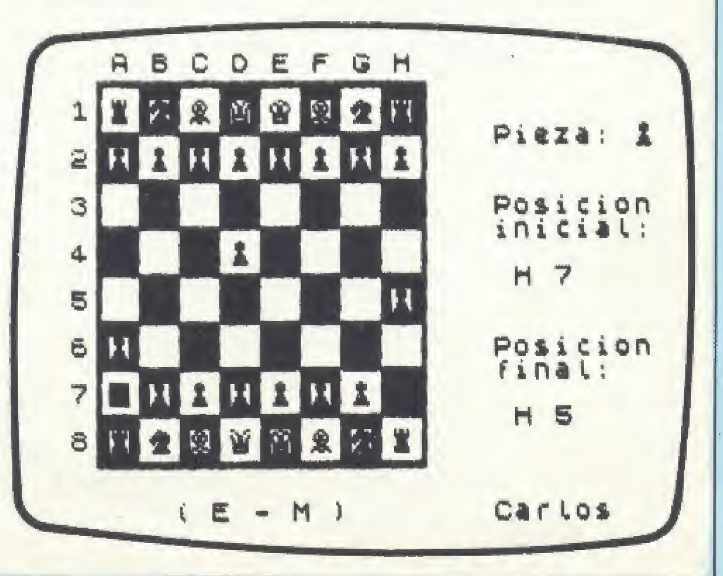
```
3. Las coordenadas de la posi-
   cion donde desea trasladar
   su pieza.

4. Para establecer las coorde-
   nadas de una pieza, pulsar
   en primer lugar la coorde-
   nada alfabetica (ABCDEFGH),
   y seguidamente la coordena-
   da numerica (12345678).

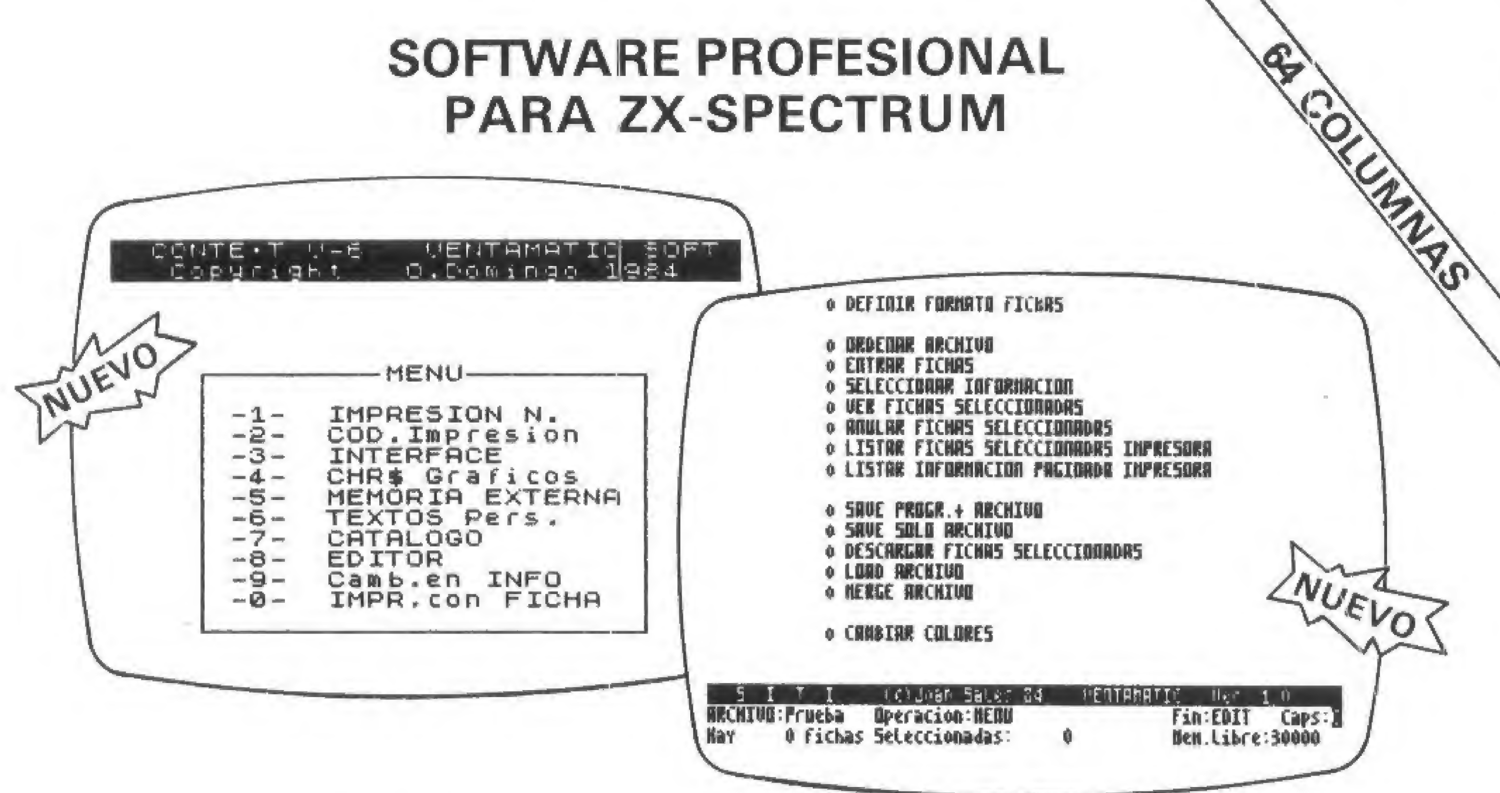
5. Presionar 'e' si ha cometi-
   do algun error. Se repetira
   la tirada.

6. Presionar 'm' si su tirada
   es correcta. Inmediatamente
   se efectuara el movimiento.

      (Presione una tecla)
```



SOFTWARE PROFESIONAL PARA ZX-SPECTRUM



CONTEXT V.6 (*) — Procesador de textos 64 columnas. Caracteres españoles. Cartas personalizadas. ¡Sensacional!

P.V.P. 4.000,— Ptas.

CONTABILIDAD PERSONAL (*) — Contabilidad doméstica o de pequeño negocio. 64 columnas. **P.V.P. 2.500,— Ptas.**

HISOF PASCAL.— El único compilador de Pascal creado para el Spectrum.

P.V.P. 6.000,— Ptas.

SITI (*) — Sistema integrado de tratamiento de información. Base de datos con posibilidad de cálculos complejos y ficheros auxiliares. **P.V.P. 4.000.— PTAS.**

BETA BASIC.— Más de 50 nuevos comandos de Basic para tu Spectrum.

P V.P. 3.000,— PTAS.

HISOF DEVPAC (*).— Conjunto de dos programas, ensamblador y desensamblador de código máquina. Impresionantes. (GENS 3M2 y MONS 3M2). **P.V.P. 3.500,— PTAS.**

CYRUS IS CHESS— El mejor ajedrez.
Campeón de Europa de 1981.
P.V.P. 3.000,— PTAS.

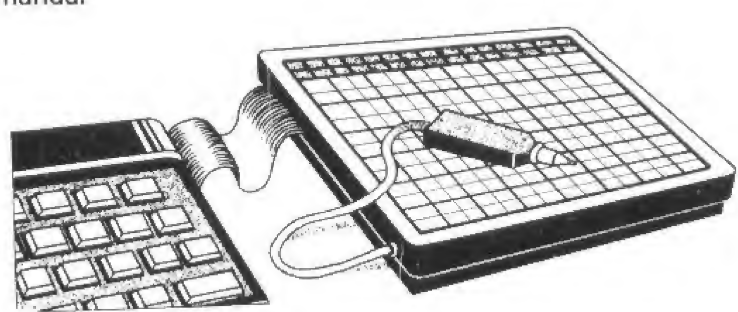
* Compatible con Microdive y/o cassette y cualquier impresora e interface de impresora

GRAFPAD

TABLETA DIGITALIZADORA PROFESIONAL PARA ZX-SPECTRUM 48 K.

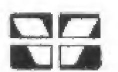
Los mejores dibujos realizados cómodamente. Trabaja hasta 3 pantallas a la vez. Incluye programa de funcionamiento y manual completo en castellano.

P.V.P. 39.000 PTAS.





TIENDA:
C/. Córcega, 89 - Entlo.
08029-BARCELONA
Tel. (91) 230 97 90

VENTAMATIC 

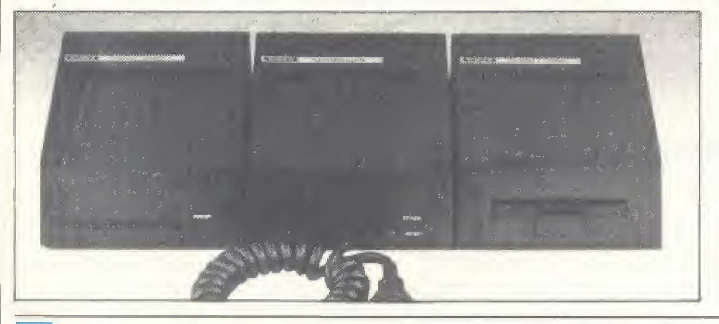
PEDIDOS POR CORREO:
Avda. de Rhode, 253
ROSAS (GERONA)
Tel. (972) 25 56 16

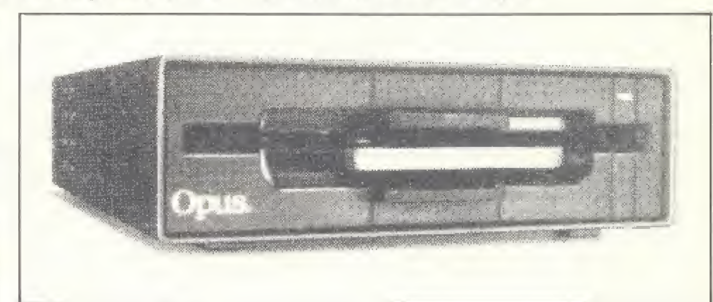
Preguntas y respuestas

P Desearía que incluyeseis en algún número próximo un artículo sobre los discos flexibles adaptables a Spectrum, sus instrucciones, capacidades y precios.

Juan Jiménez
Santander

R **Silog comercializa la unidad de discos OPUS de Technology Research Limited. Puede trabajar con discos de 100, 200 y hasta 400 Kbytes, al precio de 29.850 el interface y 55.200 el floppy. En las ferias de SONIMAG y el SIMO, Timex presentó su nueva unidad de discos, la Timex FDD System, con dos tipos de floppys, los TDD-1 de 40 sectores y 250 Kbytes y los TDD-2 de 80 sectores y 1 Megabyte. La comercialización de este floppy de Timex compatible con el QL, se realizará por Investrónica, quienes se mostraron reticentes a dar fechas y precios, más preocupados ahora en el QL.**





P Os envío unas mejoras para el programa SP-EASEL, es bastante bueno pero se puede retocar, para guardar los datos y los gráficos al grabar el programa y así poder añadir o quitar en sucesivas ocasiones nuevos datos, y para obtener el nombre de los meses en español en vez de en inglés.

```
8160 DATA "0039223a22223a000038a
0b8a0a0b800003b223b222223000 0b02
8b02828b00000293a2a2b2a2a000030a
8a8b0a8a800"
8165 DATA 5862
8170 DATA "00132a2b3a2a2b000030a
828b0a828000028a8a890909000003a0
a0a0a2a11000090a8a8a8a8280000a0a
0a0a0a03800"
8175 DATA 5963
8180 DATA "00112a2a3a2b29000010a
82828a89000001b2223120a330000b02
8a83020a00000112a2a2a2a110000b81
01010109000"
8185 DATA 5539
8190 DATA "00112a2a2a2a29000028a
8a8a89010000033292929293300000982
02020209800"
8195 DATA 3578
```

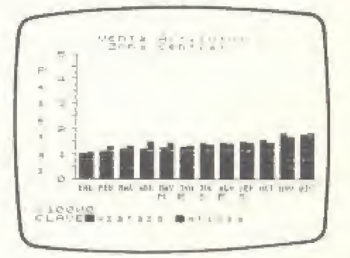
Ahora tal y como se indica en la revista número 2 se borran las líneas 8000 a 8195 después de haber corrido una vez el programa y se crea la línea:

9999 SAVE «SP-easel» LINE 10:SAVE «Easel c/m» CODE 60000, 460.

La línea 10 se cambia por: 10 LOAD "" CODE.

La línea 20 pasa a ser 55 y queda anulada. La línea 30 pasa a ser 53 y queda anulada. Y se crea la línea 52 PAPER 0: INK 7: BORDER0: CLS:GOTO500.

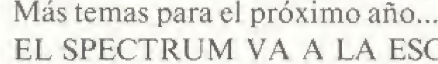
Para grabar el programa hacer GOTO 9999.



Si se borran las variables por dar RUN, hacer goto 53.

Si antes de grabar el programa se hace CLEAR 59900 se elimina el riesgo de borrar el C/M con RUN pero si se borran las variables.

R **La respuesta es sencilla: Muchas gracias por su colaboración. Permítanos obsequiarle con el número 1 que no pudo encontrar.**



Más temas para el próximo año...
EL SPECTRUM VA A LA ESCUELA. El programa más completo de representación de funciones matemáticas.
DISEÑO ASISTIDO POR ORDENADOR. La sofisticación de los grandes ordenadores, al alcance de su ordenador.
FLOPPYS PARA EL SPECTRUM. Analizamos el OPUS, ya disponible en España.
PROGRAMAS: DAMAS CHINAS, para jugar inteligentemente contra el ordenador y un SIMULADOR DE VUELO con control automático y manual, entre otros.

